Analysis of Massive Data Streams Using R

Helge Langseth[1], Anders L. Madsen[2,3],

Thomas D. Nielsen[3], Antonio Salmerón[4]

[1]Dept. Computer and Information Science. Norwegian University of Science and Technology, Trondheim, Norway

[2]Hugin Expert A/S, Aalborg, Denmark

[3]Dept. Computer Science, Aalborg University, Denmark

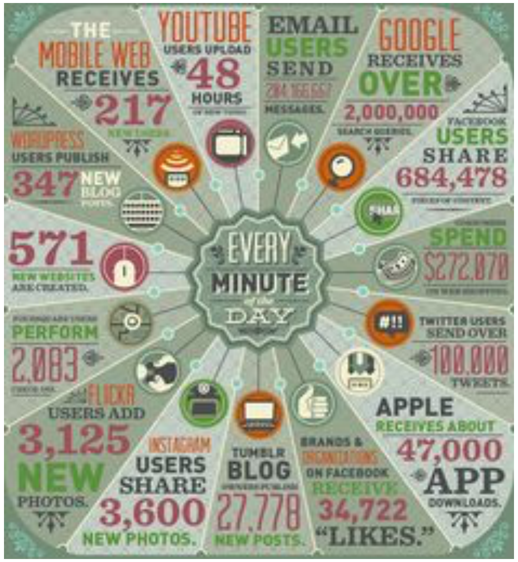
[4]Dept. Mathematics, University of Almería, Spain

Outline

1. **Introduction**
   - Data streams
   - Challenges when processing data streams
   - Why Bayesian networks?
   - The AMIDST project
2. **Bayesian networks**
   - Static and dynamic models
   - Inference and learning
3. **Exploratory analysis**
   - Exploratory time series analysis in R
   - Report generation: LaTeX + R
4. **The Ramidst package**
   - The AMIDST toolbox
   - Using the AMIDST toolbox from R

Introduction

Part I



- **Unbounded flows of data are generated daily:**
  - Social Networks
  - Network Monitoring
  - Financial/Banking industry
  - ….

Data Stream Processing

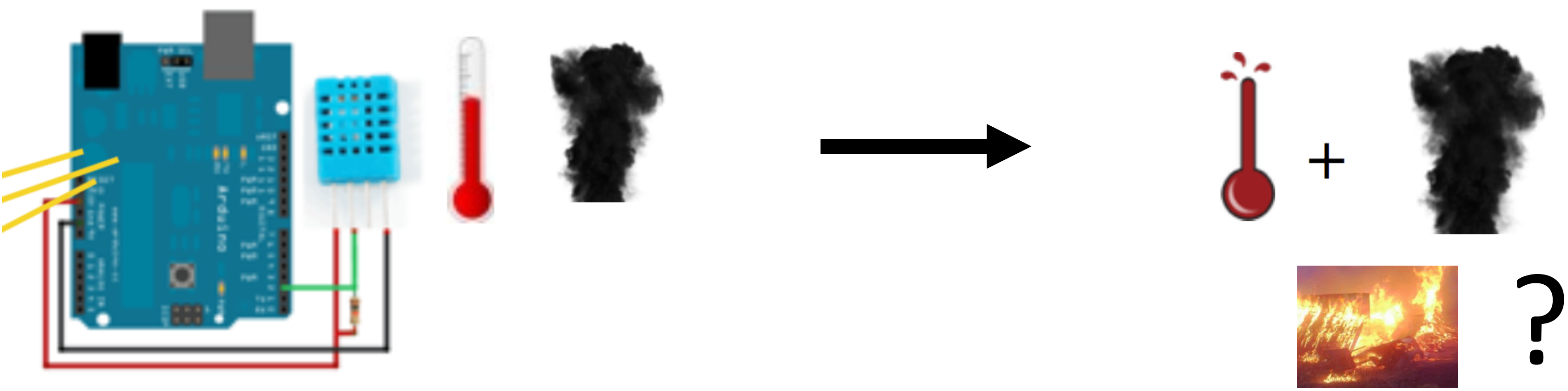
- **Processing data streams is challenging:**
  - They do not fit in main memory
  - Continuous model updating
  - Continuous inference / prediction
  - Concept drift

Processing Massive Data Streams

- **Scalability is a main issue:**
  - Scalable computing infrastructure
  - Scalable models and algorithms

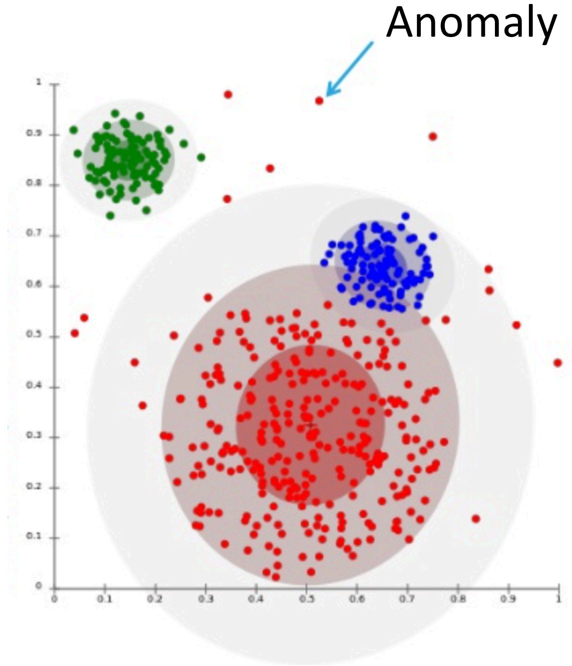
Why Bayesian networks?

- Example:
  - Stream of sensor measurements about **temperature** and **smoke** presence in a given geographical area.
  - The stream is analysed to detect the **presence of fire** (event detection problem)



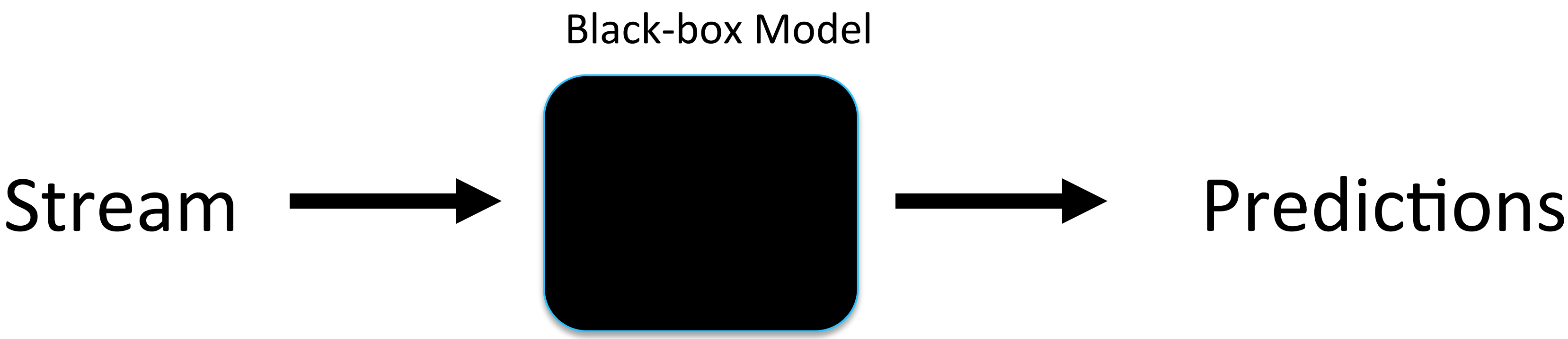
Why Bayesian networks?

- The problem can be approached as an **anomaly detection task** (outliers)
  - A commonly used method is *Streaming K-Means*



Why Bayesian networks?

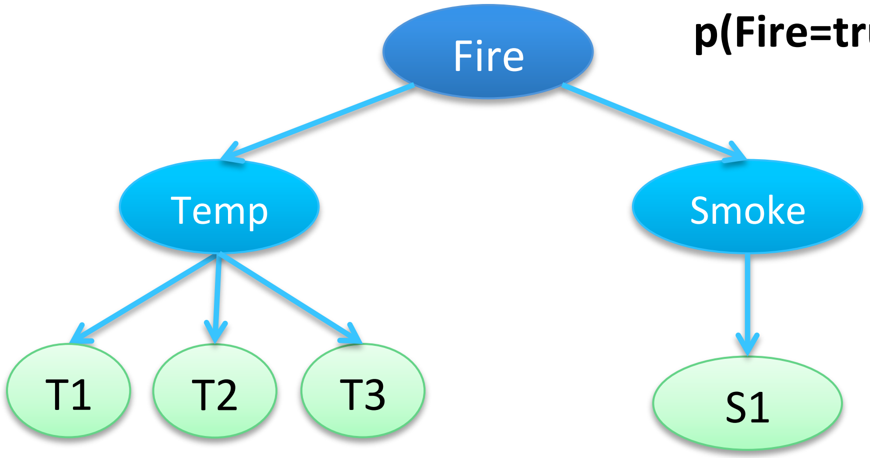
- **Often, data streams are handled using black-box models:**



- **Pros:**
  - No need to understand the problem
- **Cons:**
  - Hyper-parameters to be tuned
  - Black-box models can seldom explain away

Why Bayesian networks?

- **Bayesian Networks:**
  - Open-box models
  - Encode prior knowledge.
  - Continuous and discrete variables (CLG networks).
  - Example:

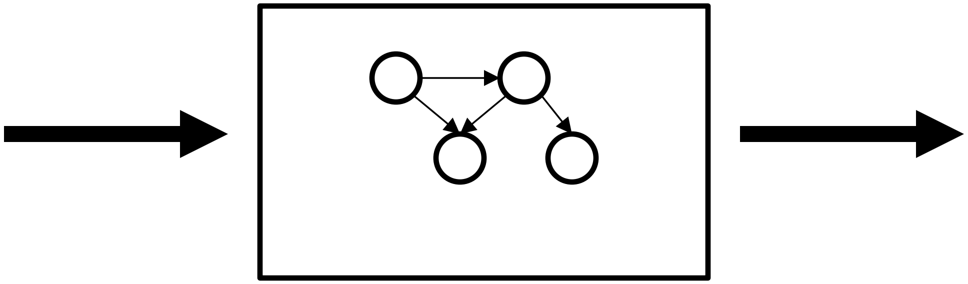


p(Fire=true|t1,t2,t3,s1)

Why Bayesian networks?

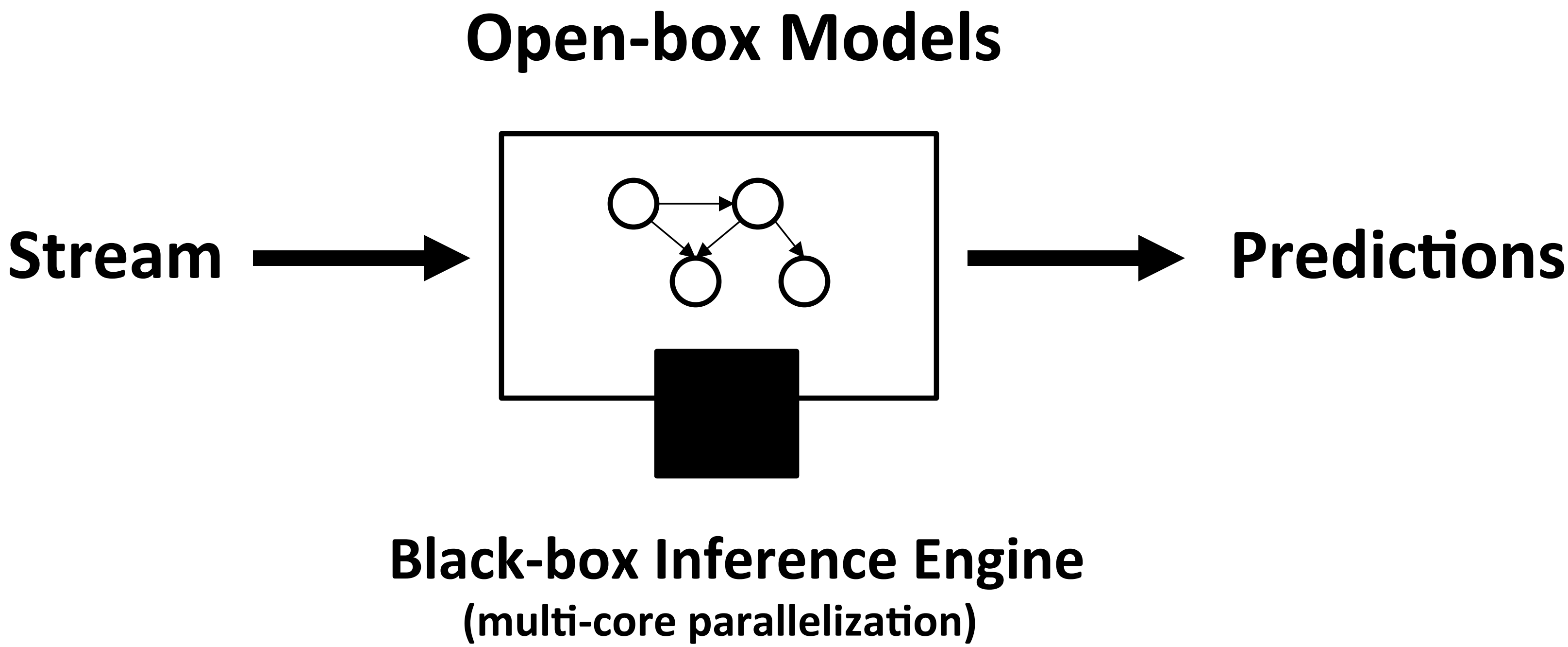
Open-box Models

Stream



Predictions

Why Bayesian networks?



The AMIDST project

- **FP7**-funded EU project
- Large number of variables
- Data arriving in streams
- Based on hybrid Bayesian networks
- Open source toolbox with learning and inference capabilities
- Two use cases provided by industrial partners
  - **Prediction of maneuvers in highway traffic (Daimler)**
  - **Risk prediction in credit operations and customer profiling (BCC)**
- **http://www.amidst.eu**

AALBORG UNIVERSITY DENMARK
BCC BANCO DE CRÉDITO COOPERATIVO
DAIMLER
HUGINEXPERT The leading decision support tool
NTNU Norwegian University of Science and Technology
UNIVERSIDAD DE ALMERÍA

Bayesian networks

Part II

Definition

- **Formally, a Bayesian network consists of**
  - A directed acyclic graph (DAG) where each node is a random variable
  - A set of conditional probability distributions, one for each variable conditional on its parents in the DAG

- **For a set of variables $\mathbf{X} = \{X_1, \ldots, X_N\}$, the joint distribution factorizes as**

$$p(\mathbf{X}) = \prod_{i=1}^{N} p(X_i | Pa(X_i))$$

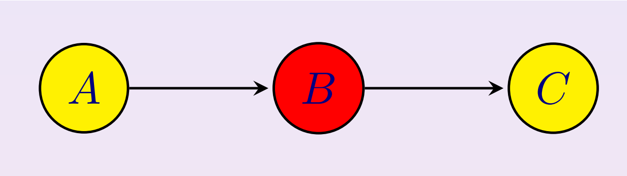
- The factorization allows local computations

Reading independencies

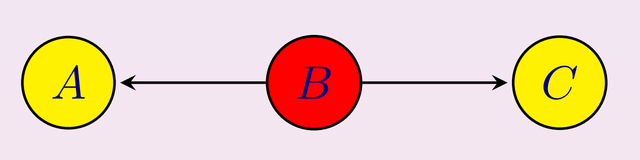
Independence relations can be read off from the structure

There are three types of connections:

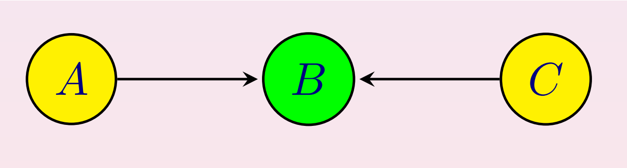
- Serial



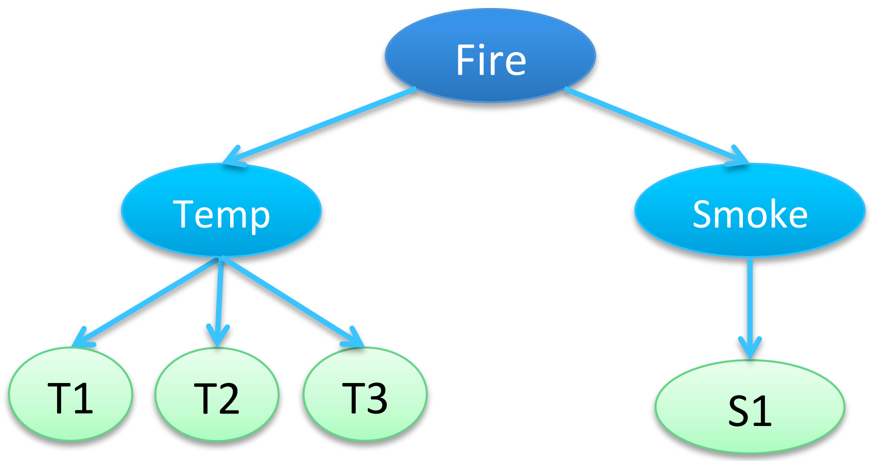
- Diverging



- Converging

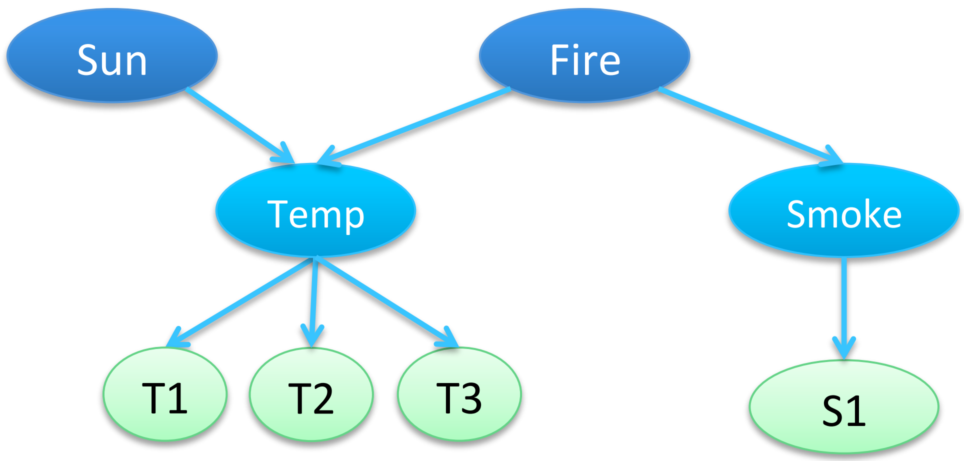


Reading independencies. Example



- Knowing the temperature with certainty makes the temperature sensor readings and the event of fire independent
- The smoke sensor reading is also irrelevant to the event of fire if Smoke is known for sure

Reading independencies. Example



- Knowing the temperature with certainty makes the temperature sensor readings and the event of fire independent
- The smoke sensor reading is also irrelevant to the event of fire if Smoke is known for sure
- If there is no any info about Temp or sensor readings, Sun and Fire are independent

Hybrid Bayesian networks

- In a hybrid Bayesian network, discrete and continuous variables coexist
- Mixtures of truncated basis functions (MoTBFs) have been successfully used in this context (Langseth et al. 2012)
    - Mixtures of truncated exponentials (MTEs)
    - Mixtures of polynomials (MoPs)

- MoTBFs support efficient inference and learning in a static setting
- Learning from streams is more problematic
- The reason is that they do not belong to the exponential family

The exponential family

- A family of probability functions belongs to the exponential family if it can be expressed as

$$f(x;\boldsymbol{\theta}) = \exp\left\{\sum_{i=1}^{k} Q_i(\boldsymbol{\theta})T_i(x) + D(\boldsymbol{\theta}) + S(x)\right\}$$

- The T_i functions are the sufficient statistics for the unknown parameters, i.e., they contain all the information in the sample that is relevant for estimating the parameters
- They have dimension 1
- We can "*compress*" all the information in the stream so far as a single number

Hybrid Bayesian networks. CLGs

A Conditional Linear Gaussian (CLG) network is a hybrid Bayesian network where

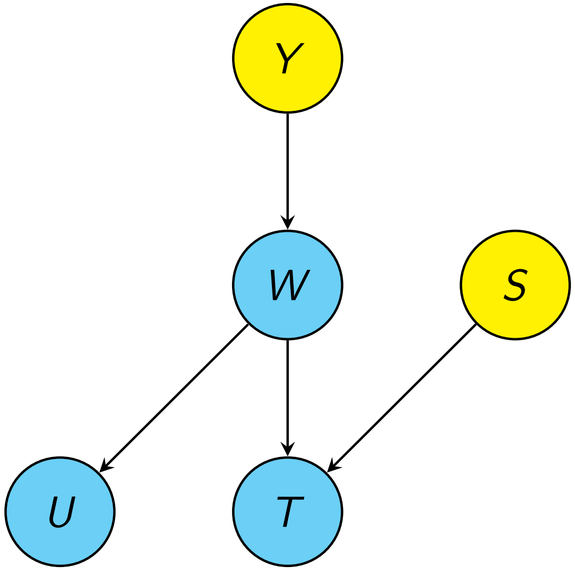
- The conditional distribution of each discrete variable X_D given its parents is a multinomial
- The conditional distribution of each continuous variable Z with discrete parents $\mathbf{X}_D$ and continuous parents $\mathbf{X}_C$, is

$$p(z|\mathbf{X}_D = \mathbf{x}_D, \mathbf{X}_C = \mathbf{x}_C) = \mathcal{N}(z; \alpha(\mathbf{x}_D) + \beta(\mathbf{x}_D)^{\mathsf{T}}\mathbf{x}_C, \sigma(\mathbf{x}_D))$$

  for all $\mathbf{x}_D$ and $\mathbf{x}_C$, where α and β are the coefficients of a linear regression model of Z given $\mathbf{X}_C$, potentially different for each configuration of $\mathbf{X}_D$.

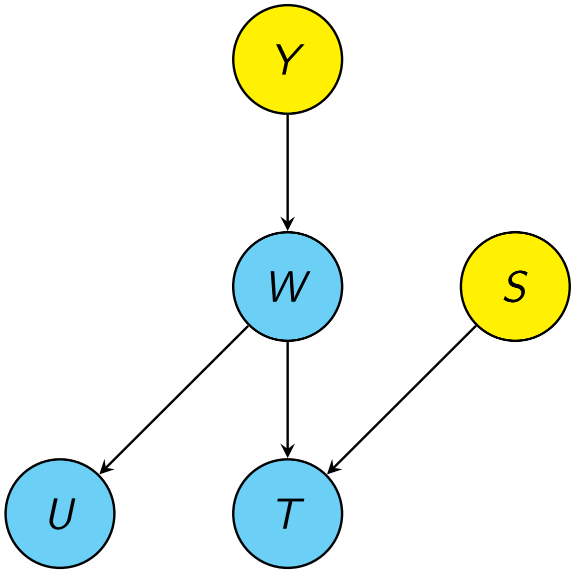
CLGs belong to the exponential family

CLGs: Example



$$P(Y) = (0.5, 0.5)$$
$$P(S) = (0.1, 0.9)$$
$$f(w|Y = 0) = \mathcal{N}(w; -1, 1)$$
$$f(w|Y = 1) = \mathcal{N}(w; 2, 1)$$
$$f(t|w, S = 0) = \mathcal{N}(t; -w, 1)$$
$$f(t|w, S = 1) = \mathcal{N}(t; w, 1)$$
$$f(u|w) = \mathcal{N}(u; w, 1)$$

CLGs: Example

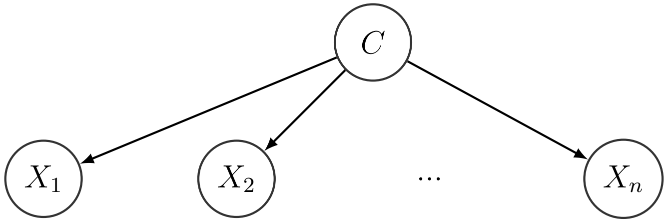


$$P(Y) = (0.5, 0.5)$$
$$P(S) = (0.1, 0.9)$$
$$f(w|Y = 0) = \mathcal{N}(w; -1, 1)$$
$$f(w|Y = 1) = \mathcal{N}(w; 2, 1)$$
$$f(t|w, S = 0) = \mathcal{N}(t; -w, 1)$$
$$f(t|w, S = 1) = \mathcal{N}(t; w, 1)$$
$$f(u|w) = \mathcal{N}(u; w, 1)$$

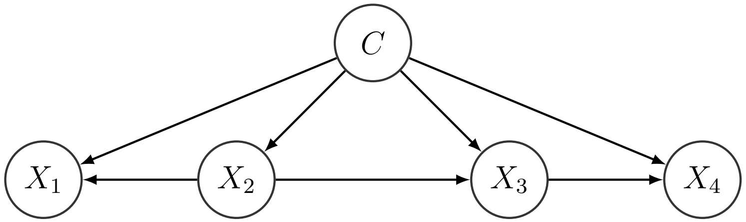
- **Limitation:** discrete nodes are not allowed to have continuous parents
- This is not a big problem for Bayesian classifiers

Bayesian network classifiers

- The structure is usually restricted
- There is a distinguished (discrete) variable called the class while the rest are called features
- Examples:



Naive Bayes



Tree-augmented network (TAN)

Bayesian network classifiers

- The class value is determined as

$$c^* = \arg\max_{c \in \Omega_C} p(c|x_1, \ldots, x_n)$$

- In the case of Naïve Bayes,

$$p(c|x_1, \ldots, x_n) \propto p(c) \prod_{i=1}^{n} p(x_i|c)$$

Reasoning over time: Dynamic Bayesian networks

- Temporal reasoning can be accommodated within BNs
- Variables are indexed over time, giving rise to dynamic Bayesian networks
- We have to model the joint distribution over time

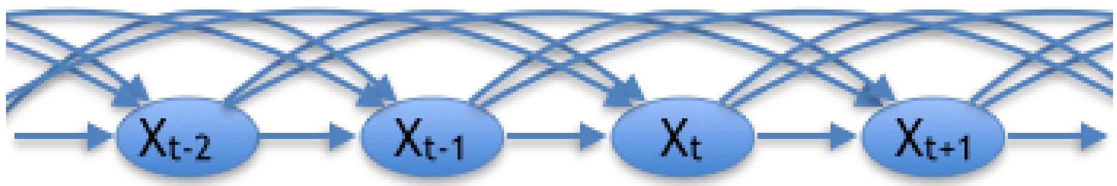
$$p(\mathbf{X}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{X}_t | \mathbf{X}_{1:t-1})$$

- Dynamic BNs reduce the factorization complexity by adopting the Markov assumption

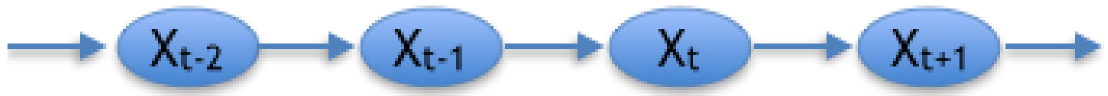
$$p(\mathbf{X}_t | \mathbf{X}_{1:t-1}) = p(\mathbf{X}_t | \mathbf{X}_{t-V:t-1})$$

Reasoning over time: Dynamic Bayesian networks

- DBN assuming third order Markov assumption

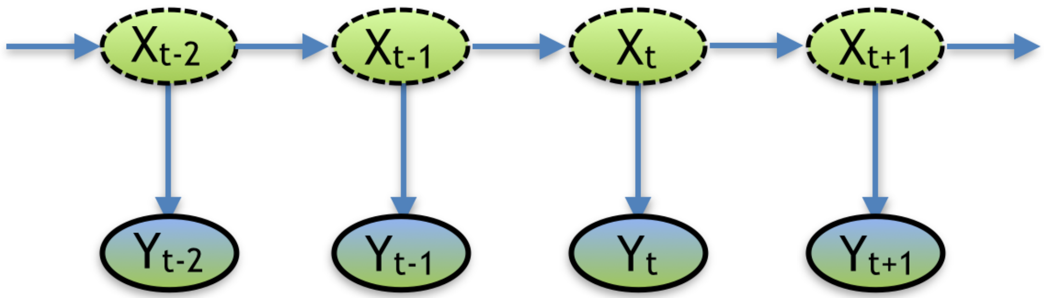


- DBN assuming first order Markov assumption



Particular cases of Dynamic Bayesian networks

- **Hidden Markov models**

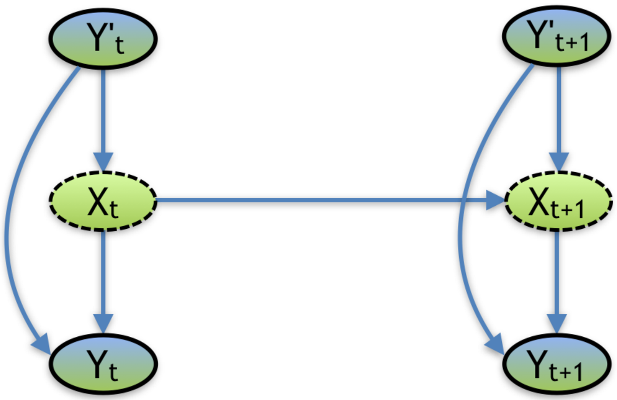


- The joint distribution of the hidden (*X*) and observed (*Y*) variables is

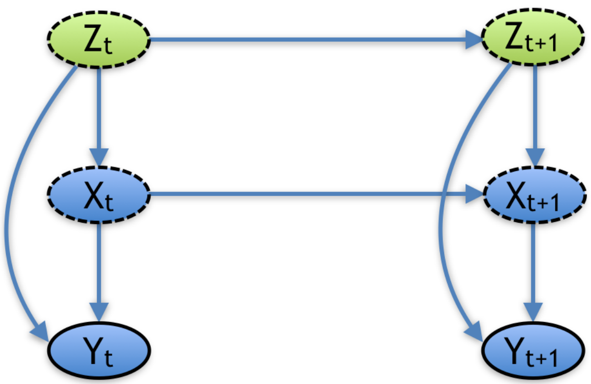
$$P(\mathbf{X}_{1:T}, \mathbf{Y}_{1:T}) = \prod_{t=1}^{t} P(\mathbf{X}_t|\mathbf{X}_{t-1})P(\mathbf{Y}_t|\mathbf{X}_t)$$

Particular cases of Dynamic Bayesian networks

- **Input-output Hidden Markov models**

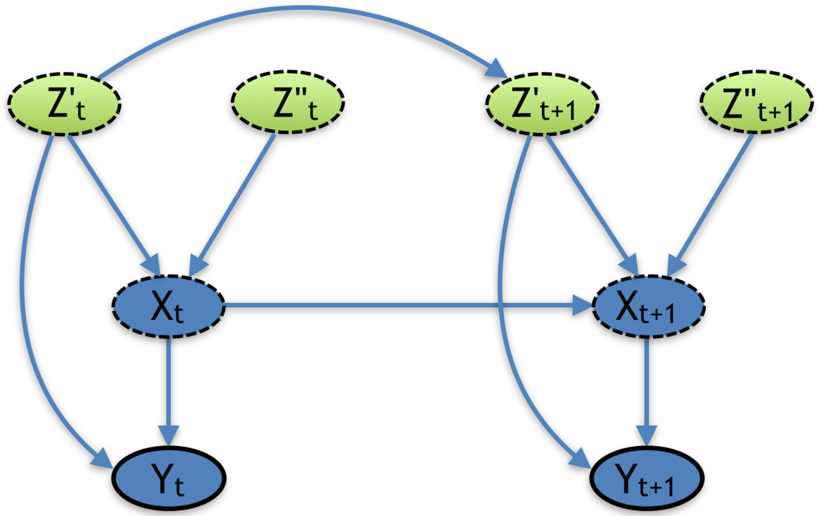


- **Linear dynamic systems: switching Kalman filter**



Two–time slice Dynamic Bayesian networks (2T–DBN)

- They conform the main dynamic model in **AMIDST**



- The **transition distribution** is

$$p(\mathbf{X}_{t+1}|\mathbf{X}_t) = \prod_{X_{t+1} \in \mathbf{X}_{t+1}} p(X_{t+1}|Pa(X_{t+1}))$$

Inference in CLG networks

- **There are three ways of querying a BN**
  - Belief updating (probability propagation)
  - Maximum a posteriori (MAP)
  - Most probable explanation (MPE)

Inference in CLG networks

- Probabilistic inference: Computing the posterior distribution of a target variable:

$$p(x_i|\mathbf{x}_E) = \frac{\displaystyle\sum_{\mathbf{x}_D} \int_{\mathbf{x}_C} p(\mathbf{x}, \mathbf{x}_E)\mathrm{d}\mathbf{x}_C}{\displaystyle\sum_{\mathbf{x}_{D_i}} \int_{\mathbf{x}_{C_i}} p(\mathbf{x}, \mathbf{x}_E)\mathrm{d}\mathbf{x}_{C_i}}$$

Inference in CLG networks

- Maximum a posteriori (MAP): For a set of target variables $\mathbf{X}_I$, the goal is to compute

$$\mathbf{x}_I^* = \arg\max_{\mathbf{x}_I} p(\mathbf{x}_I | \mathbf{X}_E = \mathbf{x}_E)$$

  where $p(\mathbf{x}_I | \mathbf{X}_E = \mathbf{x}_E)$ is obtained by first marginalizing out from $p(\mathbf{x})$ the variables not in $\mathbf{X}_I$ and not in $\mathbf{X}_E$
- Most probable explanation (MPE): A particular case of MAP where $\mathbf{X}_I$ includes all the unobserved variables

Probability propagation in CLG networks: Importance sampling

- Let's denote by θ the posterior probability for the target variable, and

$$h(x_i) = \sum_{\mathbf{x}_D \in \Omega_{\mathbf{x}_D}} \int_{\mathbf{x}_C \in \Omega_{\mathbf{x}_C}} p(\mathbf{x}; \mathbf{x}_E) \mathrm{d}\mathbf{x}_C$$

- Then,

$$\theta = \int_a^b h(x_i) \mathrm{d}x_i = \int_a^b \frac{h(x_i)}{p^*(x_i)} p^*(x_i) \mathrm{d}x_i = E_{p^*}\left[\frac{h(X_i^*)}{p^*(X_i^*)}\right]$$

Therefore, we have transformed the problem of probability propagation into estimating the expected value of a random variable from a sample drawn from a distribution of our own choice

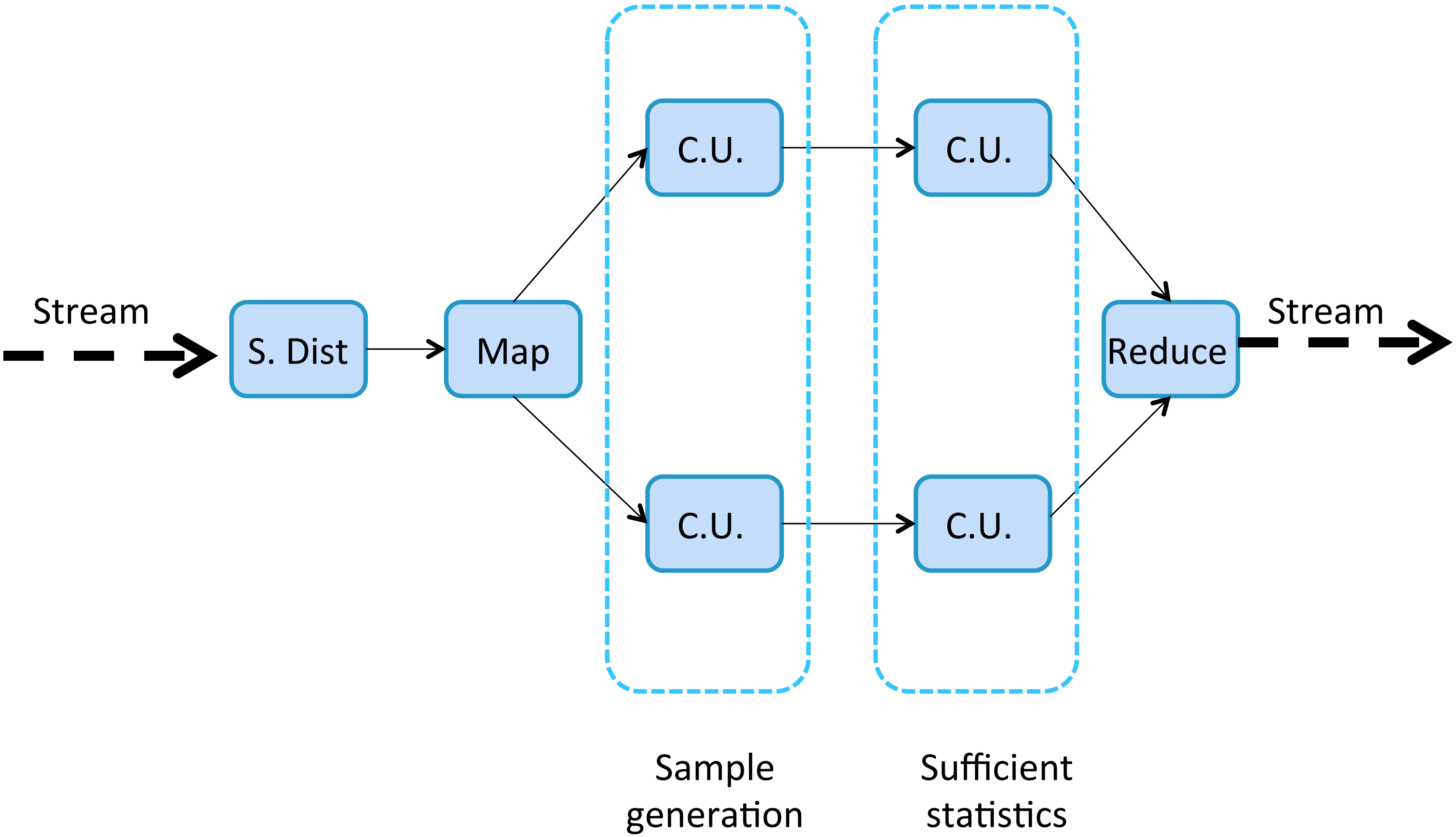
Probability propagation in CLG networks: Importance sampling

- The expected value can be estimated using a sample mean estimator. Let $X_i^{*(1)}, \ldots, X_i^{*(m)}$ be a sample drawn from *p**. Then a consistent unbiased estimator of θ is given by

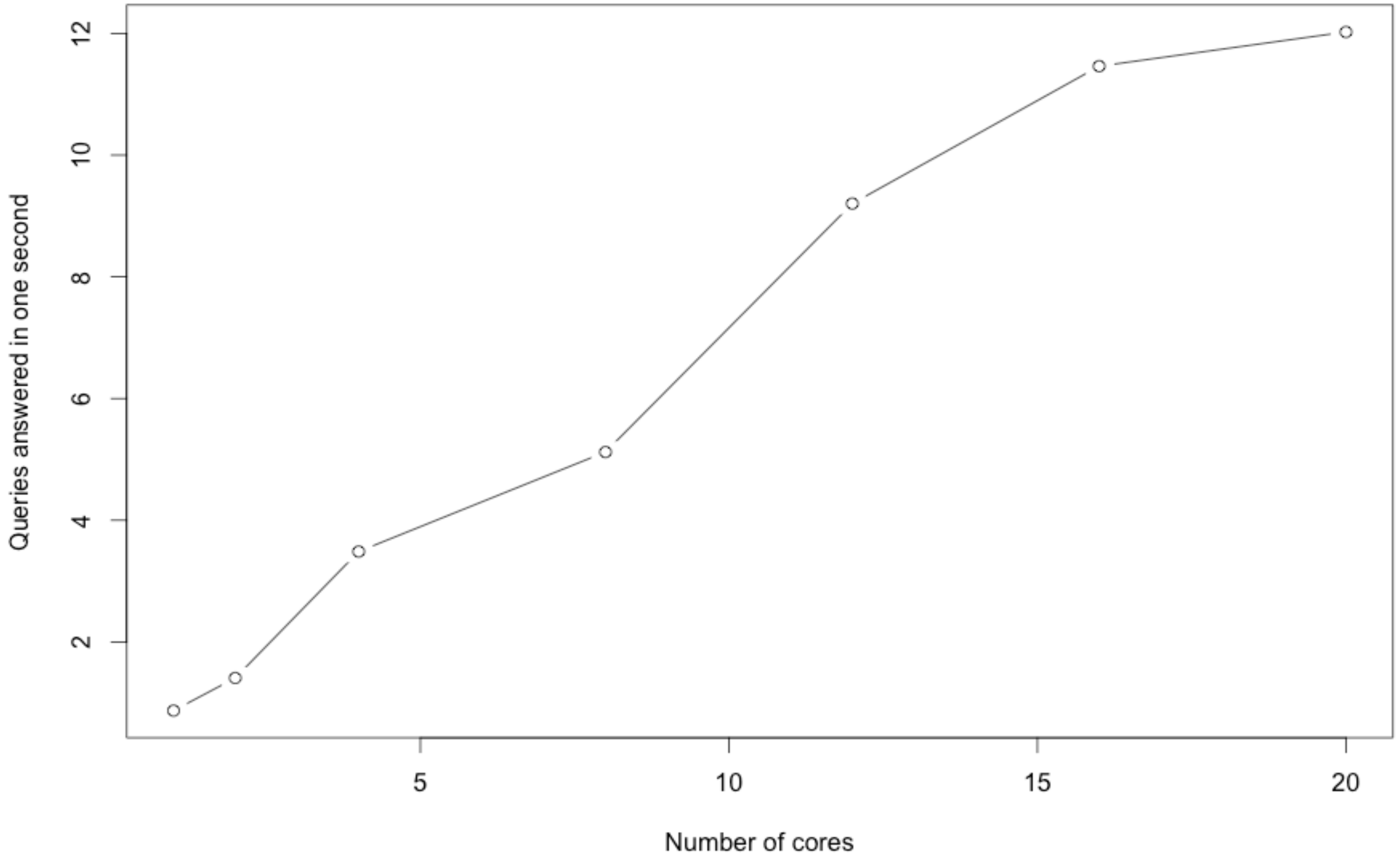
$$\hat{\theta}_1 = \frac{1}{m} \sum_{j=1}^{m} \frac{h(X_i^{*(j)})}{p^*(X_i^{*(j)})}$$

- In AMIDST, the sampling distribution is formed by the conditional distributions in the network (Evidence weighting)

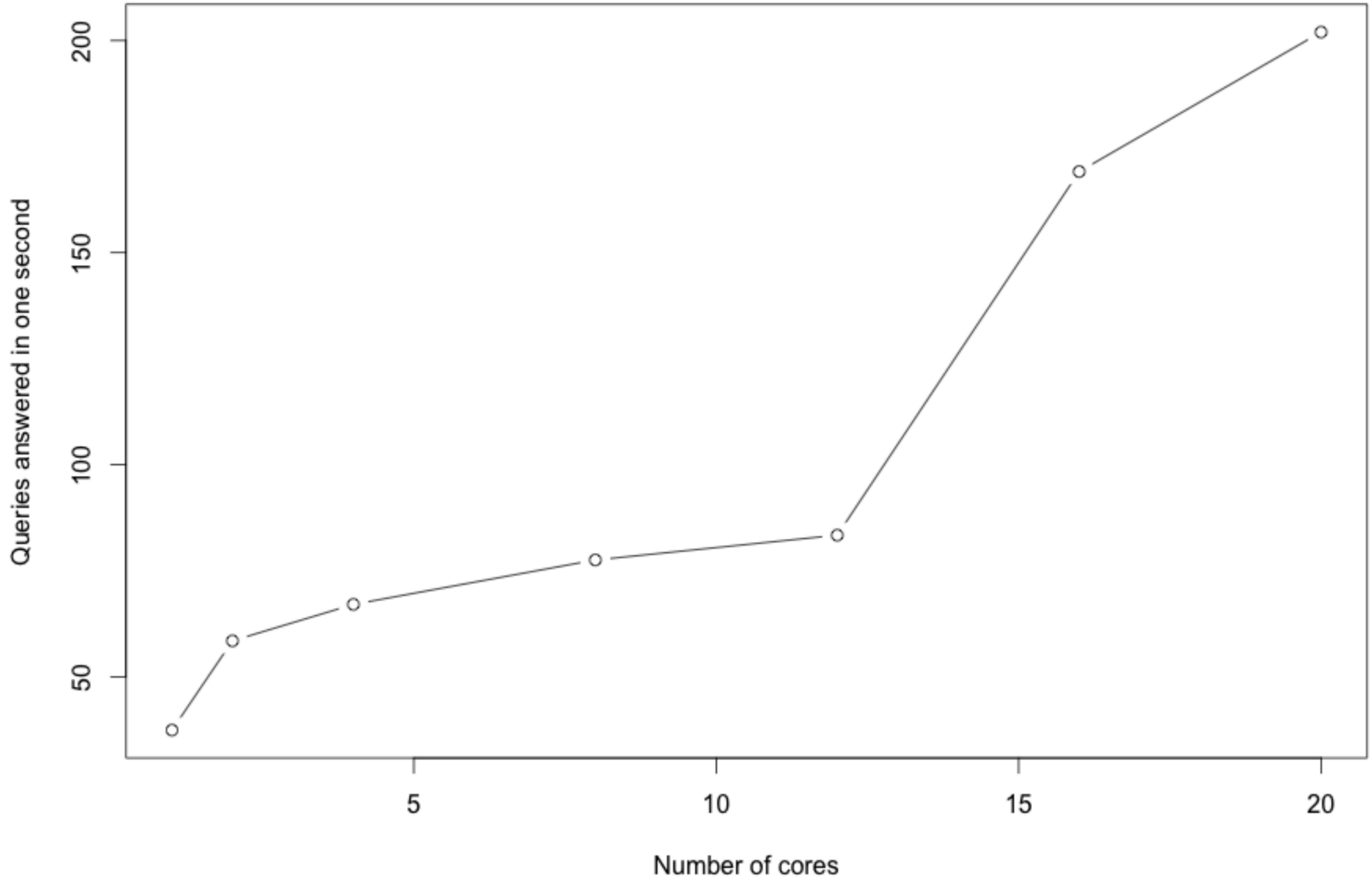
Probability propagation in CLG networks: Importance sampling



Probability propagation in CLG networks: Importance sampling



Probability propagation in CLG networks: Importance sampling



MAP in CLG networks

MAP is similar to probability propagation but:

- First marginalize out by sum/integral (sum phase)
- Then maximize (max phase)

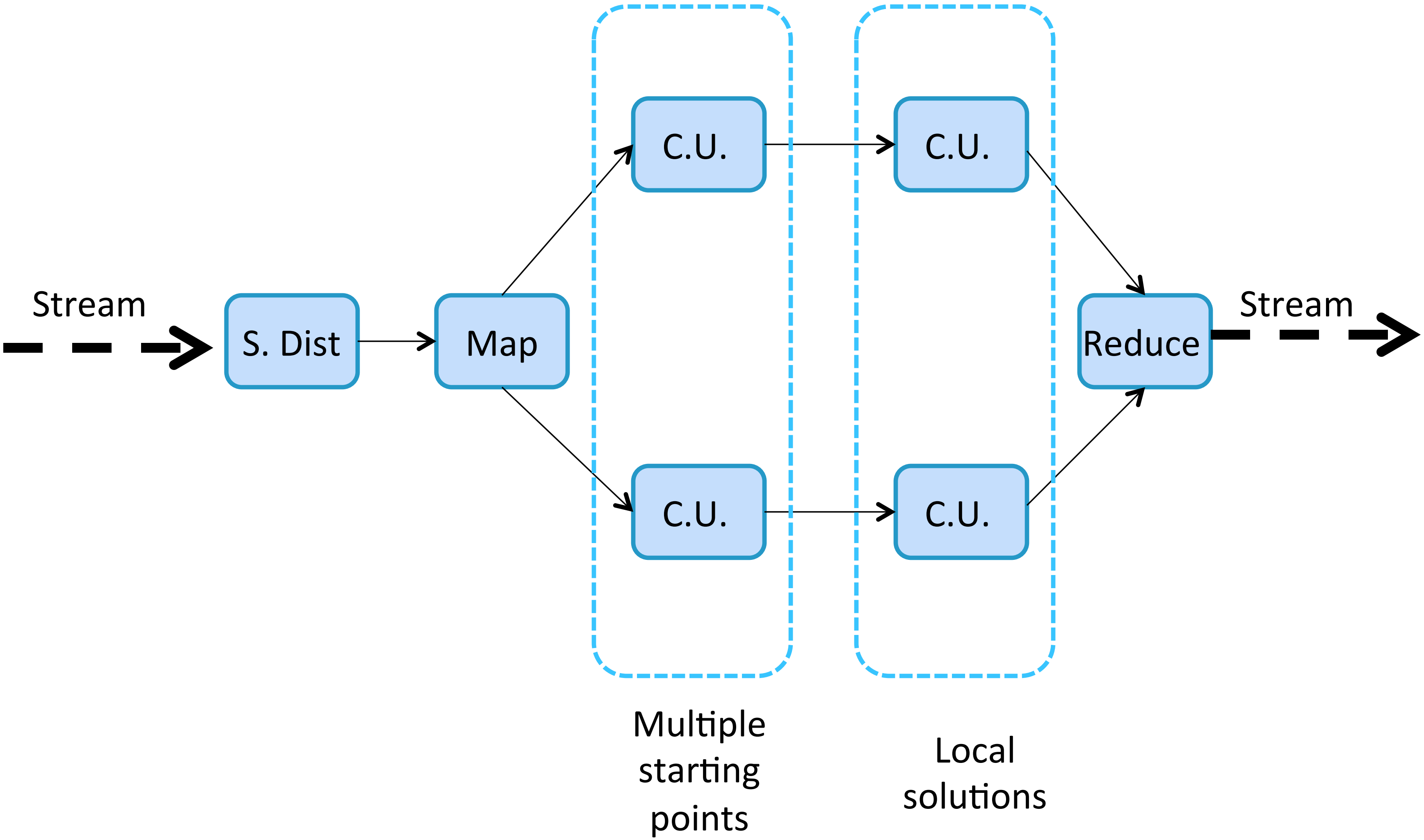
Constrained order -> higher complexity

MAP in CLG networks

MAP in the AMIDST Toolbox

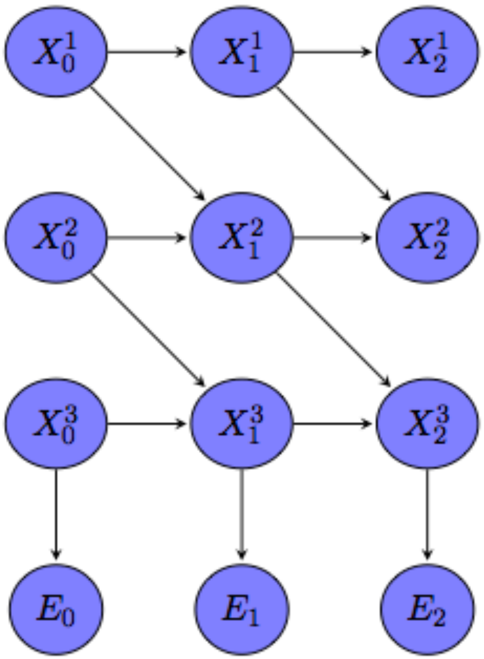
- Hill Climbing (global and local change)
- Simulated Annealing (global and local change)
- Sampling

MAP in CLG networks



Inference in dynamic networks

Inference in DBNs faces the problem of entanglement:



All variables used to encode the belief state at time $t = 2$ become dependent after observing $\{e_0, e_1, e_2\}$.

Inference in dynamic networks

- Variational message passing based on the variational approximation to a posterior distribution $p(\mathrm{x}_I)$ which is defined as

$$q^*(\mathbf{x}_I) = \arg\min_{q \in \mathcal{Q}} D(q(\mathbf{x}_I) || p(\mathbf{x}_I))$$

- Factored frontier, which assumes independence of the nodes connecting to the past given the observations

Learning CLG networks from data

- **Learning the structure**
  - Methods based on conditional independence tests
  - Score based techniques
- **Estimating the parameters**
  - Bayesian approach
  - Frequentist approach (maximum likelihood)

Learning CLG networks from data

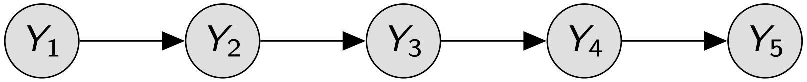
- **Bayesian parameter learning**
  - Parameters are considered random variables rather than fixed quantities
  - A prior distribution is assigned to the parameters, representing the state of knowledge before observing the data
  - The prior is updated in the light of new data.
  - The Bayesian framework naturally deals with data streams

$$p(\theta|d_1,\ldots,d_n,d_{n+1}) \propto p(d_{n+1}|\theta)p(\theta|d_1,\ldots,d_n)$$

Learning CLG networks from data

Simple example:

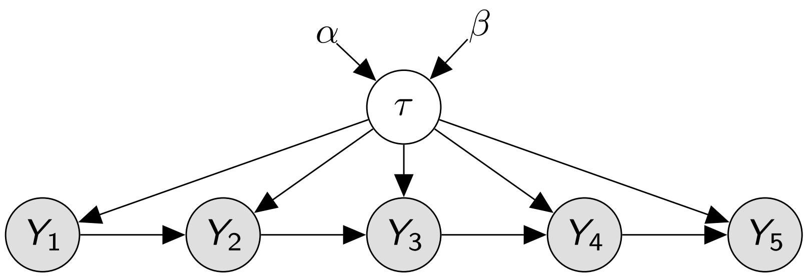
- Random walk over $Y_1, Y_2, \ldots$
- $f(y_t|y_{t-1}) \sim N(y_{t-1}, \tau^{-1})$.
- Precision τ is unknown.



Learning CLG networks from data

Simple example:

- Random walk over $Y_1, Y_2, \ldots$
- $f(y_t | y_{t-1}) \sim N(y_{t-1}, 1/\tau)$.
- Precision τ is unknown.



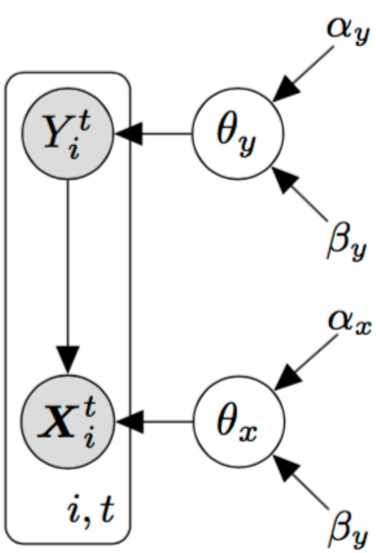
The Bayesian solution:

- Model unknown parameters as random variables.
- Use Bayes formula with "clever" distribution families:

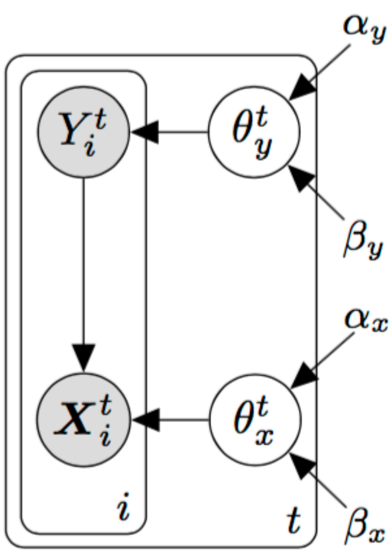
$$f(\tau | y_{1:T}, a, b) = \frac{f(\tau | a, b) \prod_{t=1}^{T} f(y_t | y_{t-1}, \tau)}{f(y_{1:T} | a, b)}.$$

Efficient inference leads to efficient learning!

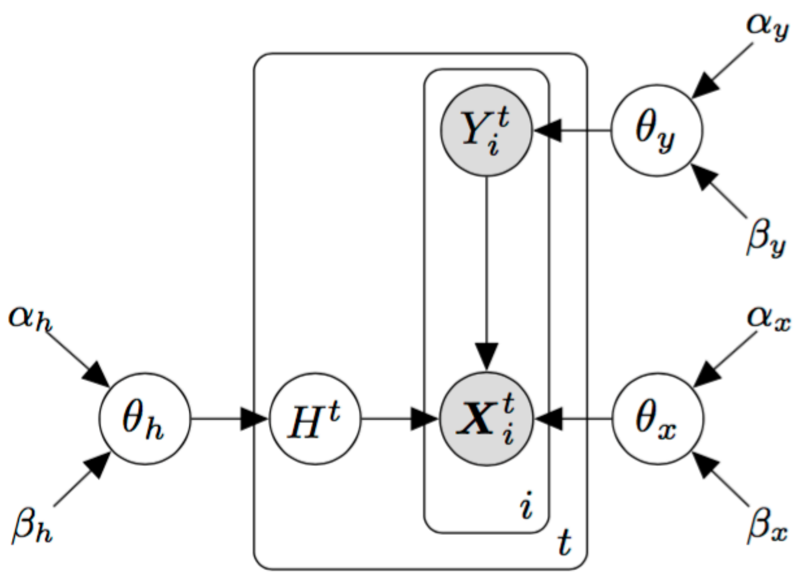
Modeling concept drift with DBNs



(a) No concept drift

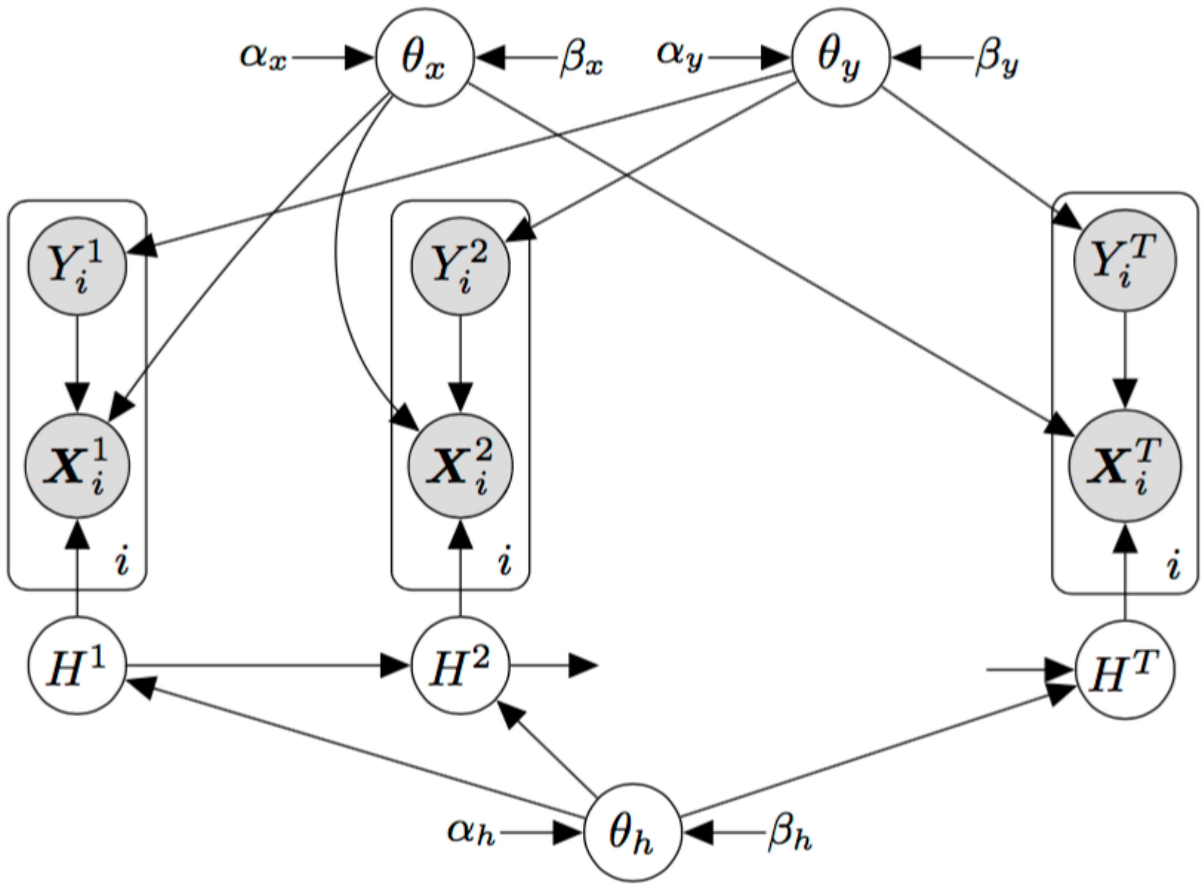


(b) Concept drift



(c) Modeling concept drift with a latent variable

Modeling concept drift with DBNs



Exploratory analysis

Part III

Exploratory analysis

- Exploratory analysis helps us in testing model assumptions
- It also improves the modeler's knowledge about the problem and its nature
- Dynamic Bayesian networks aim at modeling complex time correlations

Sample correlogram

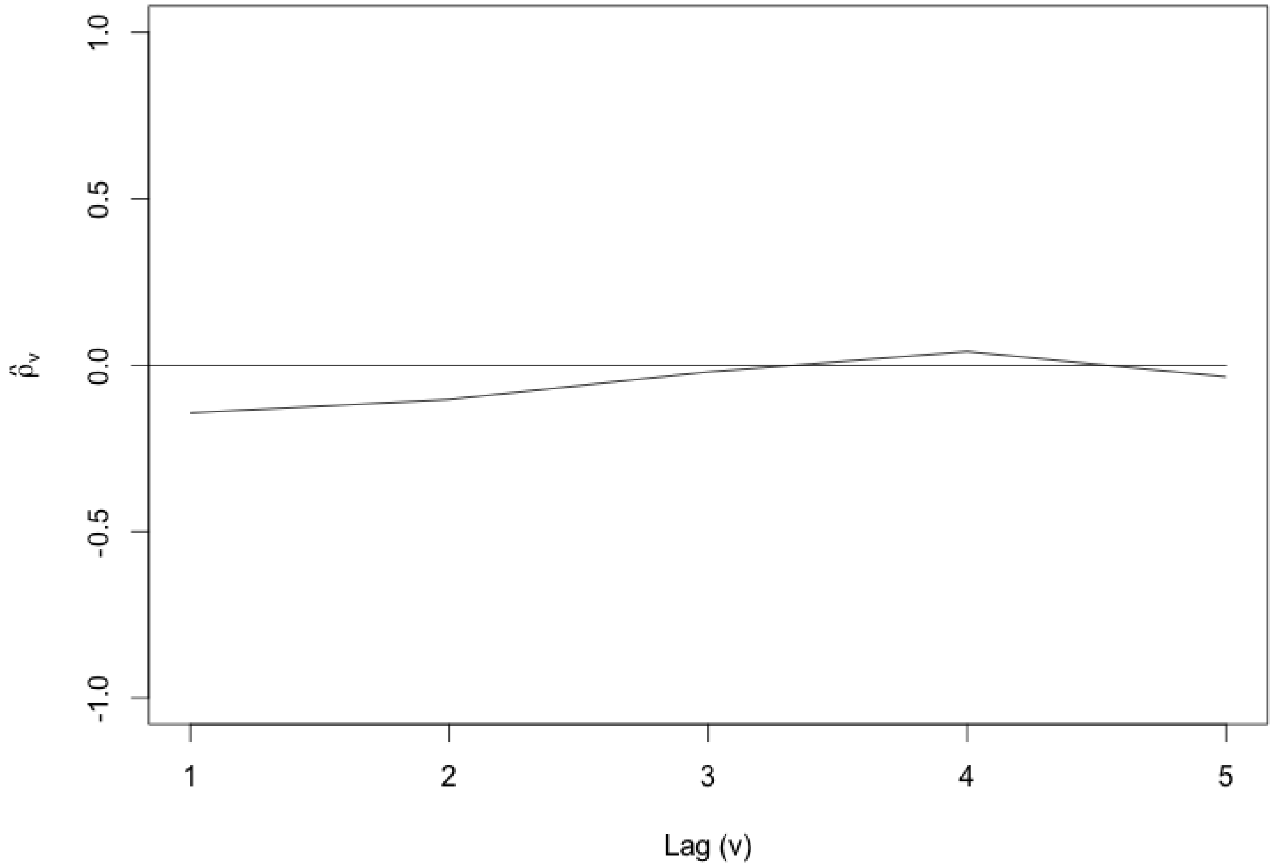
- Let $x_1,\ldots,x_T$ be a univariate time series. The sample autocorrelation coefficient at lag v is given by

$$\hat{\rho}_v = \frac{\sum_{t=1}^{T-v}(x_t - \bar{x})(x_{t+v} - \bar{x})}{\sum_{t=1}^{T}(x_t - \bar{x})^2}$$

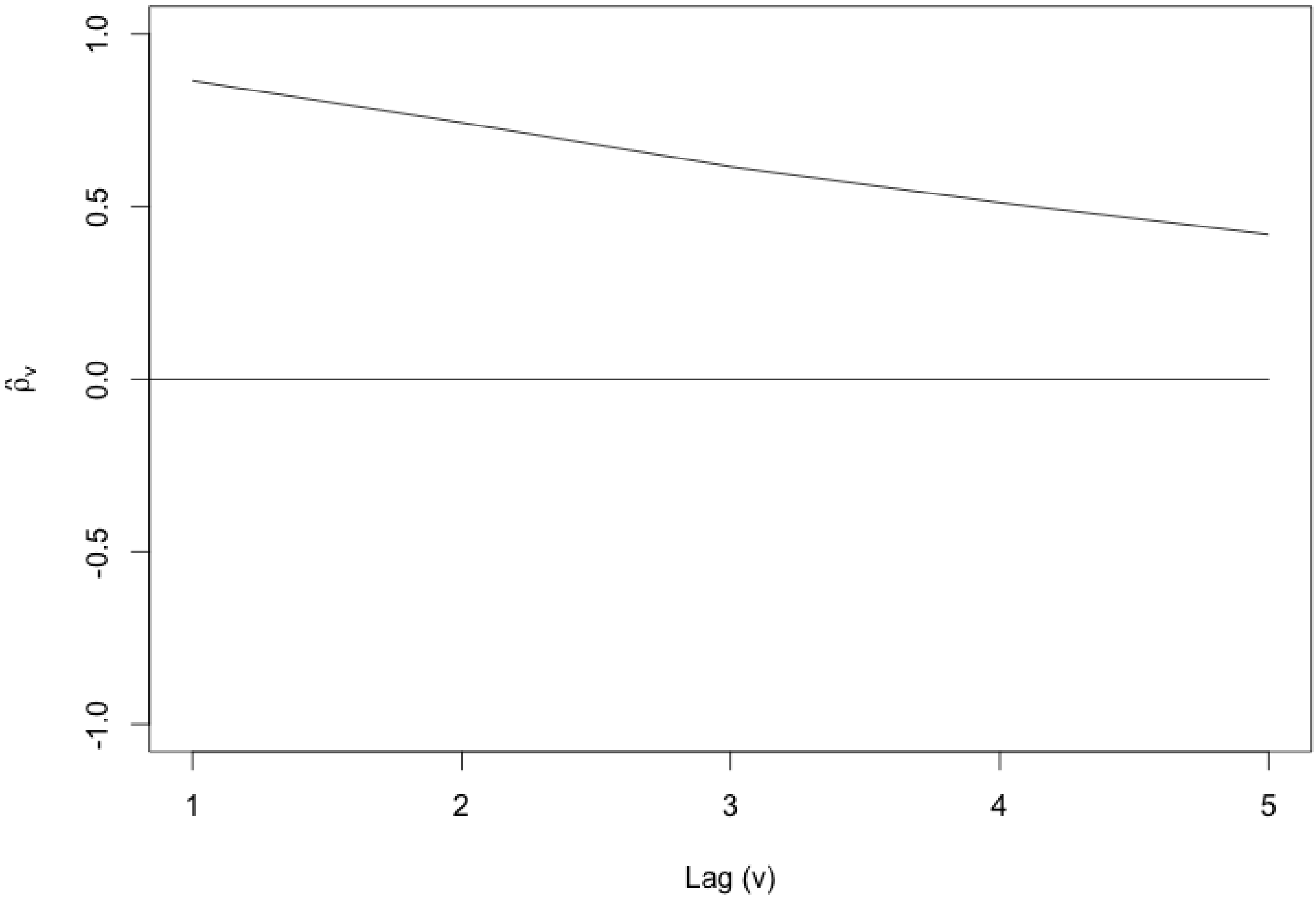
- It represents Pearson's correlation coefficient between series $\{x_t\}_{t\in\{1,\ldots,T\}}$ and $\{x_{t+v}\}_{t+v\in\{1,\ldots,T\}}$

The sample correlogram is the plot of the sample autocorrelation vs. v

Sample correlogram for independent data



Sample correlogram for time correlated data



Sample partial correlogram

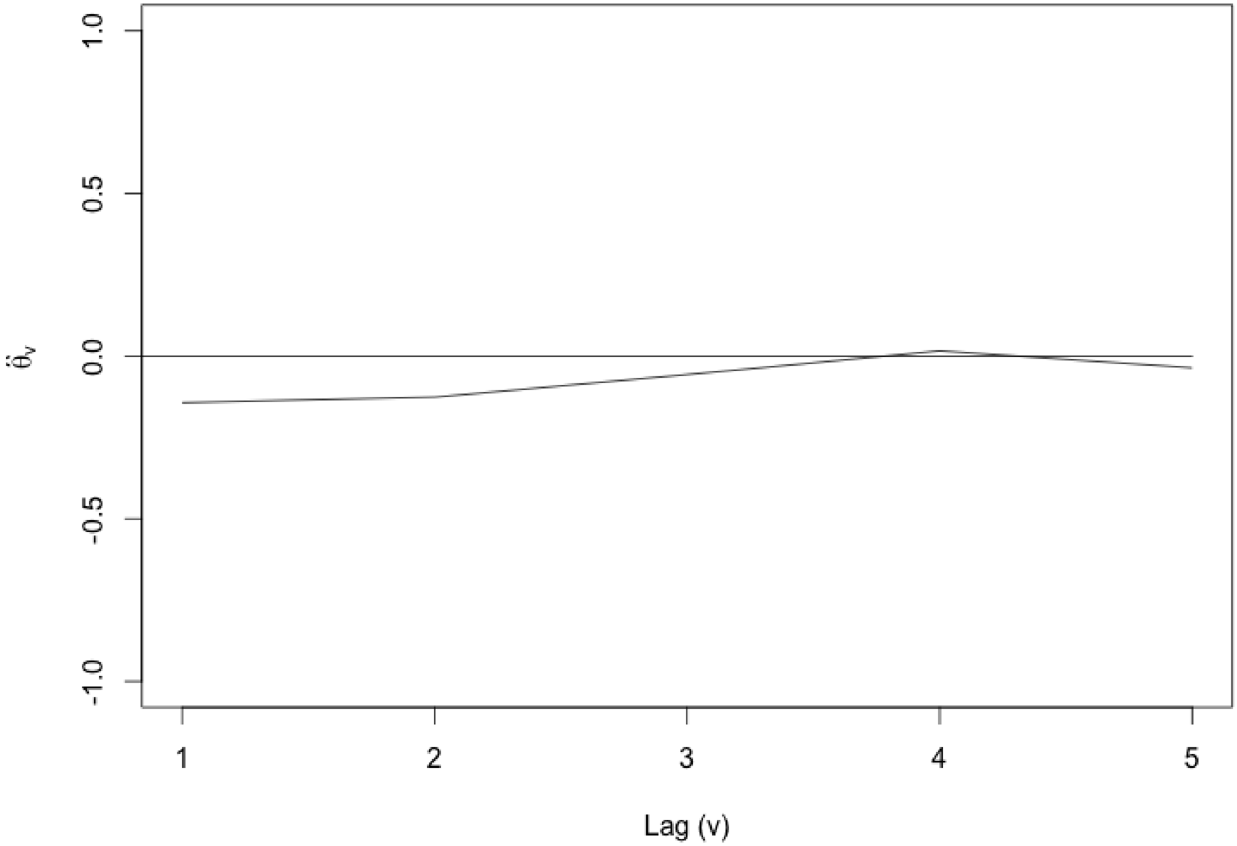
- Consider the regression model

$$X_t = a_0 + a_1 X_{t-1} + a_2 X_{t-2} + \ldots a_{v-1} X_{t-v+1}$$

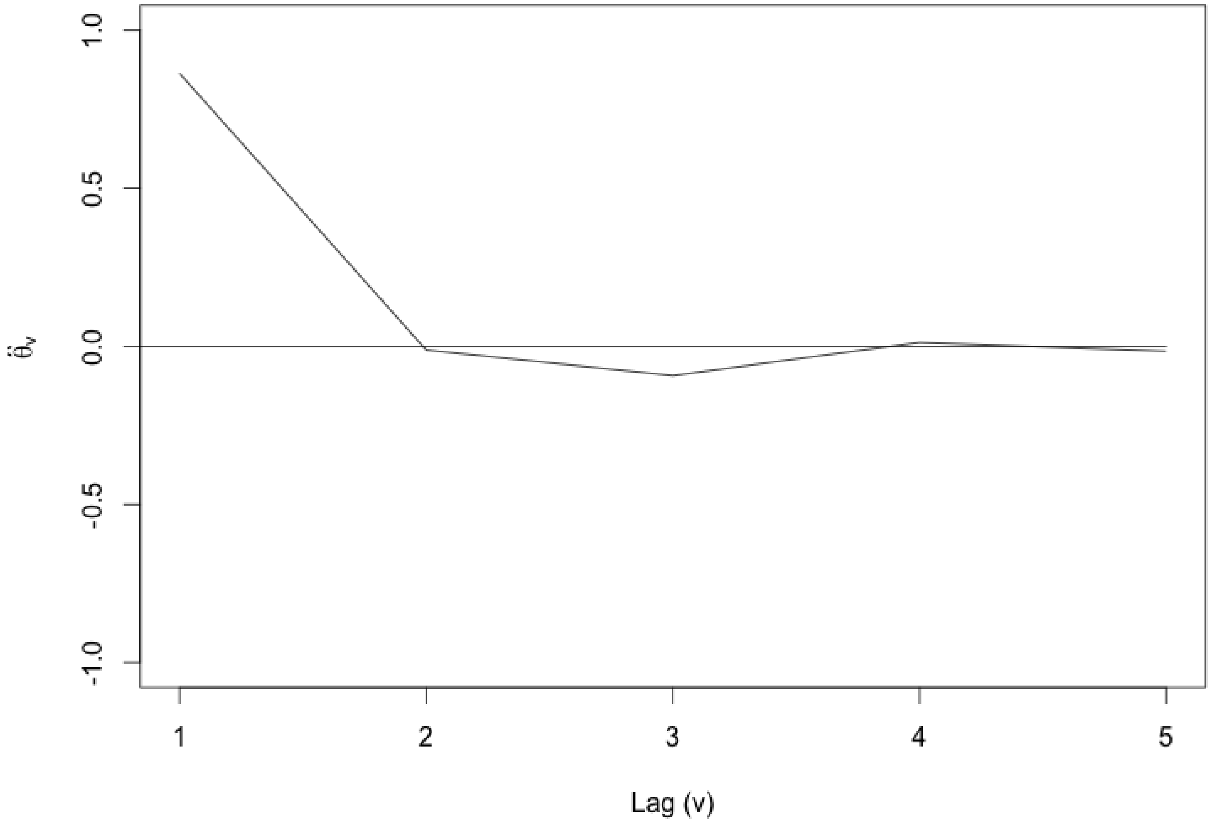
- Let $e_{t,v}$ denote the residuals
- The sample partial auto-correlation coefficient of lag v is the standard sample auto-correlation between the series $\{x_{t-v}\}_{t-v \in \{1,\ldots,T\}}$ and $\{e_{t,v}\}_{t \in \{1,\ldots,T\}}$

- It can be seen as the correlation between X_t and X_{t-v} after having removed the common linear effect of the data in between.

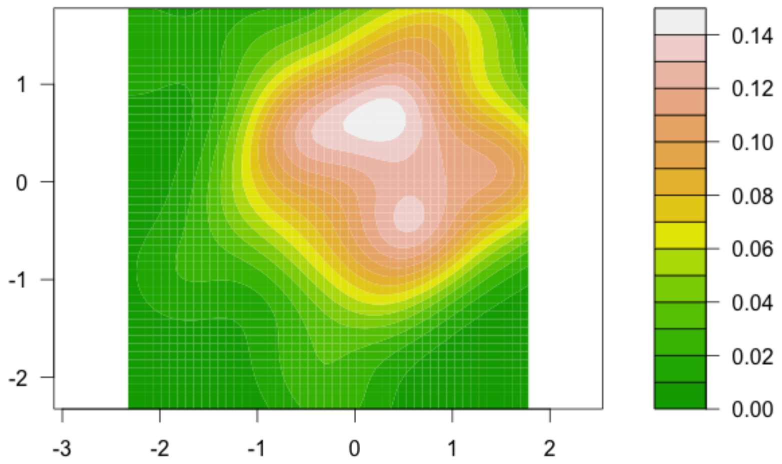
Sample partial correlogram for independent data



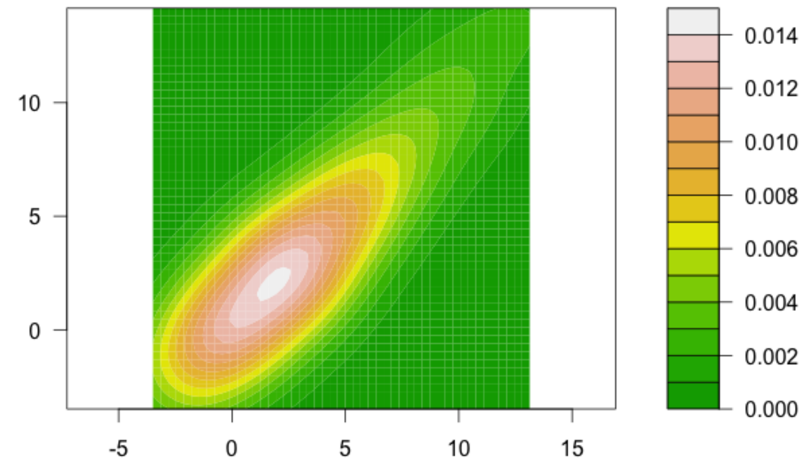
Sample partial correlogram for time correlated data



Bivariate contour plots



(a) i.i.d. data



(b) Time series data

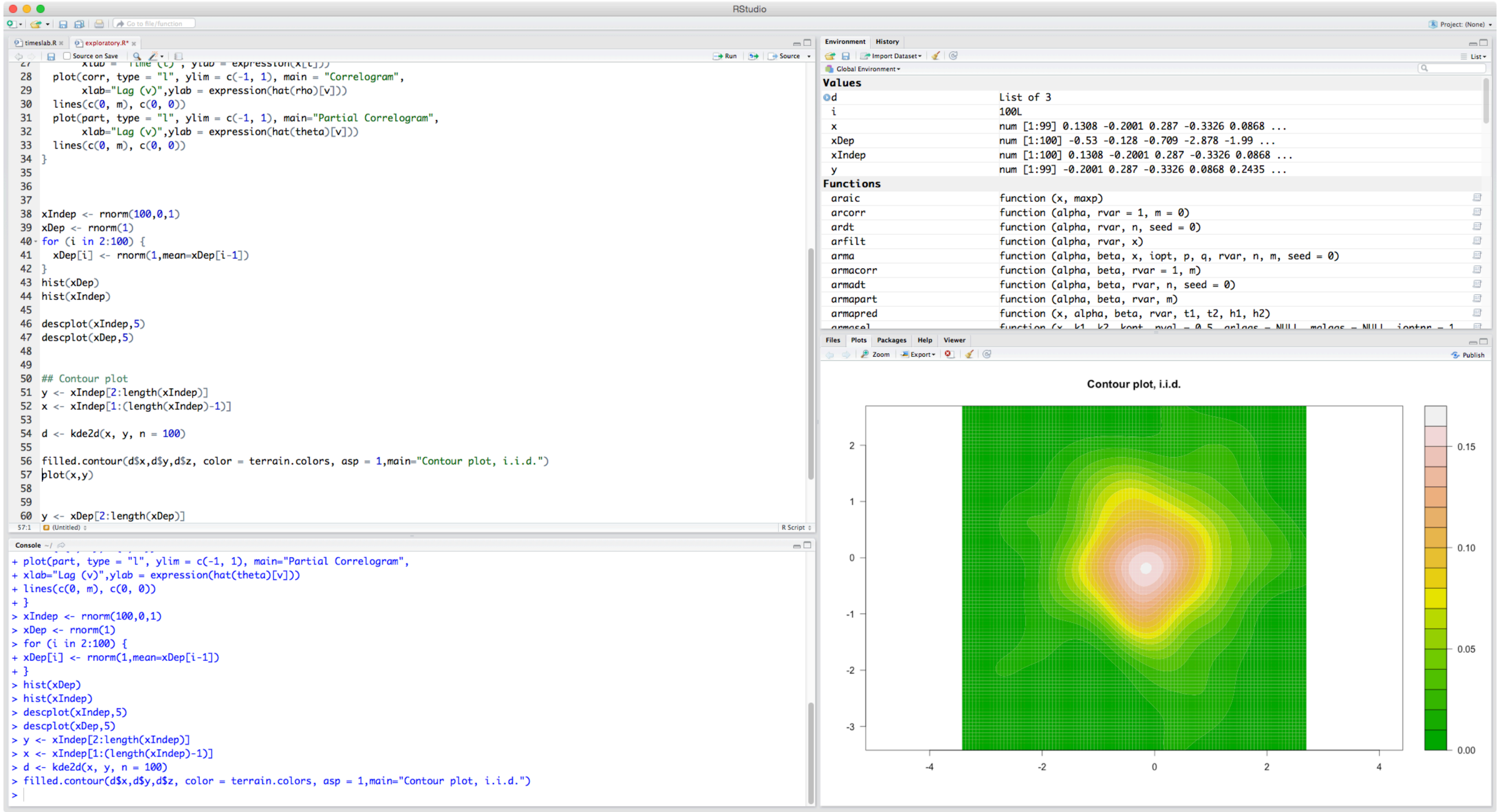
The R statistical software

- R has become a successful tool for data analysis
- Well known in Statistics, Machine Learning and Data Science communities
- “Free software environment for statistical computing and graphics”

http://www.cran.r-project.org

The Rstudio IDE

http://www.rstudio.com



The R statistical software

- Exploratory analysis demo using R
- Latex document generation from R using Sweave

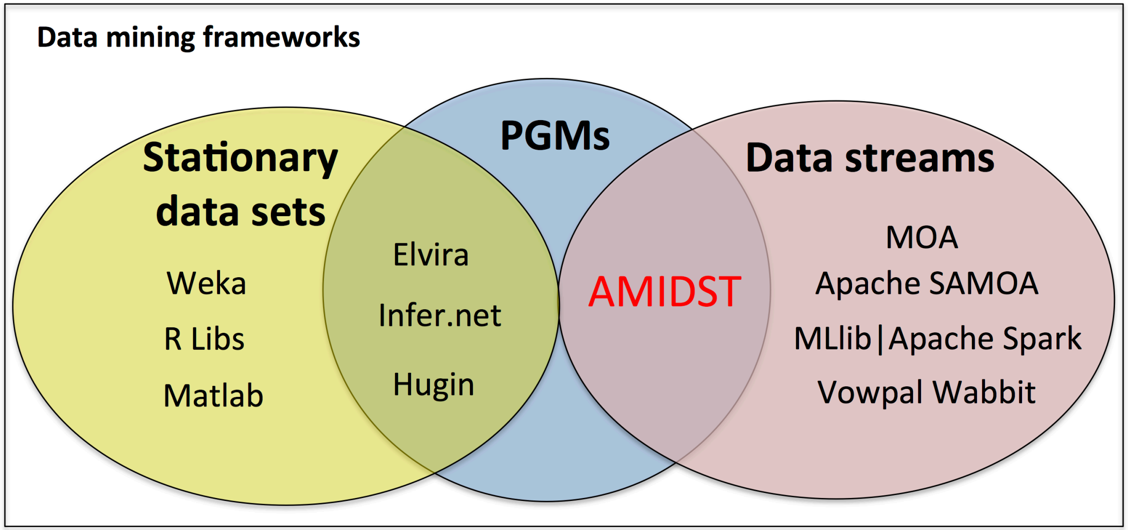
The Ramidst package

Part IV

The Ramidst package

- The package provides an interface for using the AMIDST toolbox functionality from R
- The interaction is actually carried out through the rJava package
- So far Ramidst provides functions for inference in static networks and concept drift detection using DBNs
- Extensive extra functionality available thanks to the HUGIN link

The AMIDST toolbox



- **Scalable framework for data stream processing.**
  - Based on Probabilistic Graphical Models.
  - Unique FP7 project for data stream mining using PGMs.
  - Open source software (Apache Software License 2.0).

The AMIDST toolbox official website

Scope

This toolbox offers a collection of scalable and parallel algorithms for inference and learning of hybrid Bayesian networks (BNs) from streaming data. For example, AMIDST provides parallel multi-core implementations of Bayesian parameter learning, using streaming variational Bayes and variational message passing. Additionally, AMIDST efficiently leverages existing functionalities and algorithms by interfacing to existing software tools such as Hugin and MOA. AMIDST is an open source Java toolbox released under the Apache Software License version 2.0.

http://amidst.github.io/toolbox/

Available for download at GitHub

- Download:

  :> git clone https://github.com/amidst/toolbox.git

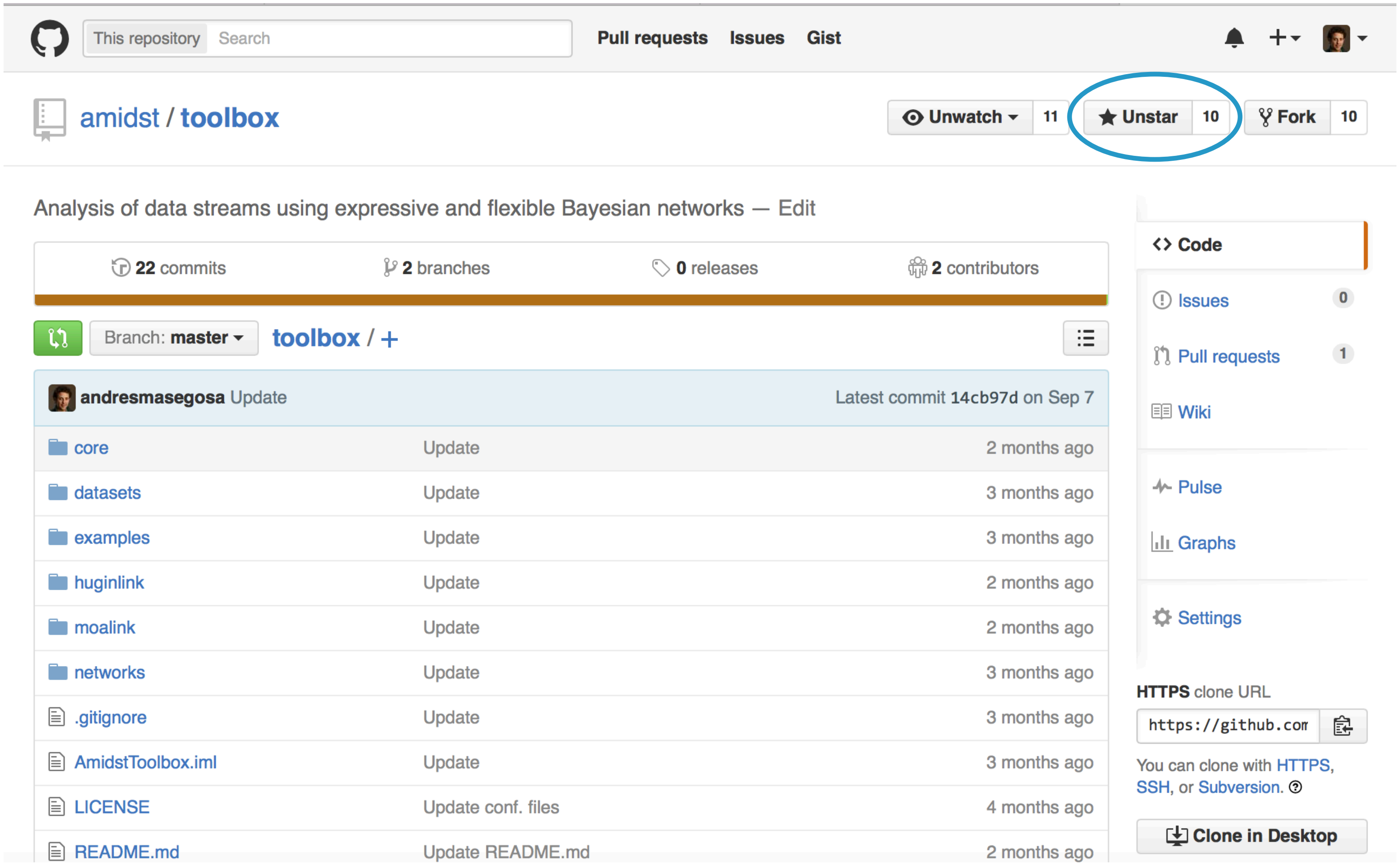
- Compile:

  :> ./compile.sh

- Run:

  :> ./run.sh <class-name>

Please give our project a “star”!



Processing data streams in R

- **RMOA**
  - MOA is a state-of-the-art tool for data stream mining.
  - RMOA provides functionality for accessing MOA from R
  - Several static models are available
  - They can be learnt from streams
  - Streams can be created from csv files or from different R objects

http://moa.cms.waikato.ac.nz

The Ramidst package

Inference and concept drift demo
using Ramidst

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 619209