CDN - An R Package for Cumulative Distribution Networks

MAZO, Gildas et PHAM, Van Trung

Inria Grenoble Rhône-Alpes
Laboratoire Jean Kuntzmann (LJK)

Cumulative Distribution Networks (CDN's)

What is a **CDN**?

- A distribution function (df) which is a product of bivariate df's.
- A graph encoding dependencies between variables.

How to do **inference**?

- To compute the likelihood is not possible by hand.
- A message passing algorithm (MPA) [1] exists to do that.
- Implementation is complicated: may prevent users from using CDN's.

Our work: implement MPA to render inference easier.

Example.

Let x_1, x_2, x_3 be some random variables of interest and θ the unknown parameter vector. The CDN writes as

$$F(x_1, x_2, x_3, \theta) = \Phi_1(x_1, x_2, \theta)\Phi_2(x_2, x_3, \theta),$$

where Φ_1 and Φ_2 are bivariate df's. The MPA aims to compute:

$$\frac{\partial^3 F}{\partial x_1 \partial x_2 \partial x_3}(x_1, x_2, x_3, \theta) \quad \text{(likelihood)}$$

$$\nabla_\theta \frac{\partial^3 F}{\partial x_1 \partial x_2 \partial x_3}(x_1, x_2, x_3, \theta) \quad \text{(likelihood gradient).}$$

The tree corresponding to this CDN is represented figure 1.

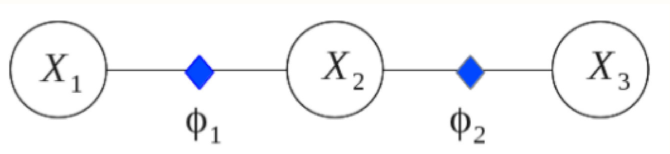

Figure 1: An example of a three variables CDN

Package description

The user specifies

- the **edges** between the variables in the graph,
- the **family** of the bivariate df's (the Φ's) by
  - specifying himself a handmade df, or
  - choosing a pre-existing model (e.g. Gaussian, Gumbel, ...) (much faster).

The package provides tools to

- differentiate symbolically the bivariate df's given by the user (if any) and store them in a file ,
- **create a CDN object** containing the inputs needed by MPA consisting (among others) in
  - the tree adjacent matrix,
  - the bivariate functions and their derivatives;
- **implement and launch in C++ the MPA algorithm** to compute the CDN likelihood and the gradient,
- **get the results back in R**.

Dependencies: packages "igraph" and "Rcpp".

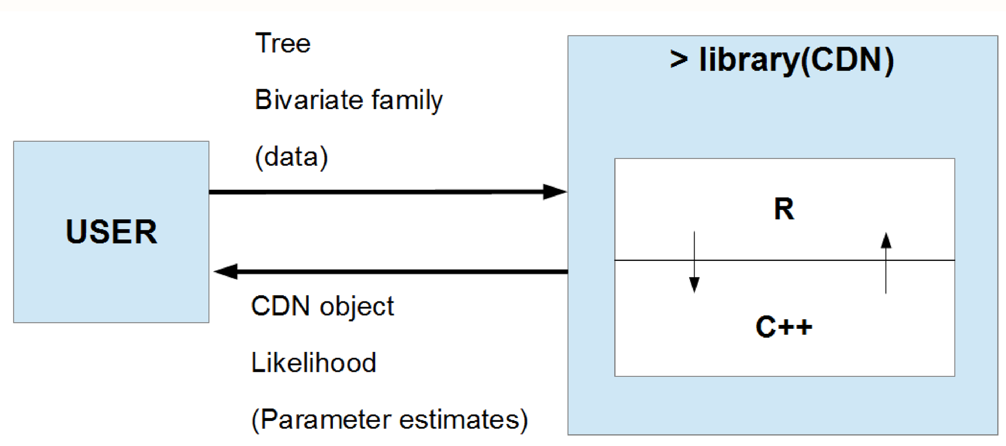

Figure 2: CDN package description.

An example

- 104 monthly **sea level measurements** at 19 sites in Japan from 2001 to 2011 are analysed.
- Tree graph based on **geographic proximity** is set up.
- Bivariate Gumbel logistic distributions are chosen.
- Parameters are estimated.

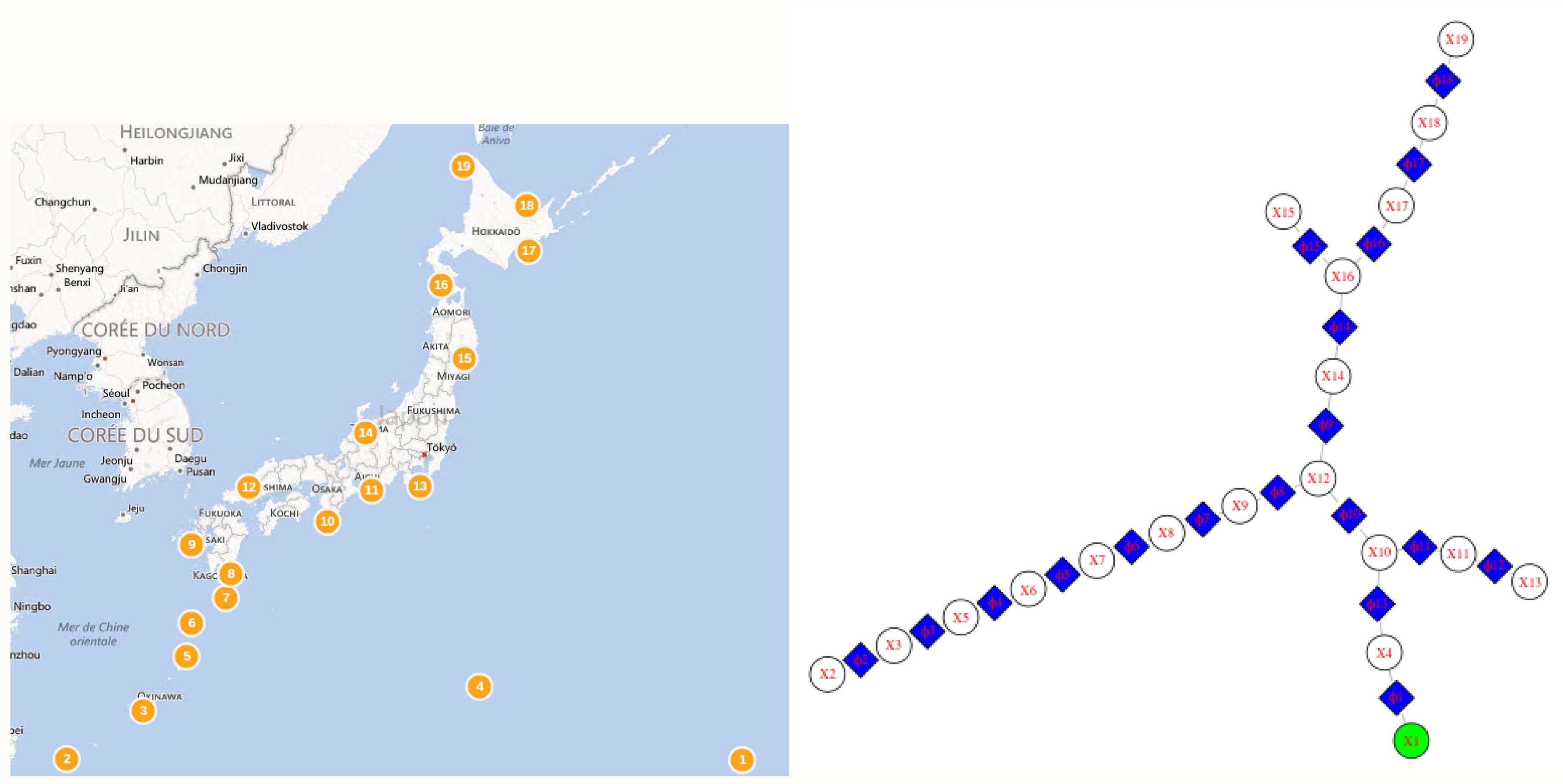

Figure 3: The 19 sites in Japan and the chosen tree.

R code

```
> # The tree is built:
> g <- graph.formula(X1-X4, X2-X3, X3-X5, X5-X6, X6-X7, X7-X8, X8-X9,
X9-X12, X12-X14, X12-X10, X10-X11, X11-X13, X10-X4, X14-X16, X16-X15,
X16-X17, X17-X18, X18-X19, simplify = FALSE)
> # The user provides a handmade bivariate df family:
> f<-expression(exp(-(x^(-1/theta)+y^(-1/theta))^(theta)))
> # Or chooses a pre-existing model:
> model <- "Gumbel"
> # A CDN object is created:
> CDN <- cdn(g, root="X1", distribution = f)
> CDN <- cdn(g, root="X1", model = "Gumbel")
> # The tree can be plotted
> cdnPlot(CDN)
> # Density (likelihood) and gradient can be computed:
> CDNout <- cdnCompute(CDN, x = rexp(19),theta = runif(18), ...)
> CDNout$density # density
> CDNout$gradient # gradient
> # The likelihood can be maximized:
> MLfit1 <- optim(par, fn = function(theta){cdnCompute(theta,...)$density},
gr = function(theta){cdnCompute(theta,...)$gradient})
> MLfit2 <- cdnTraining(CDN, data)
```

Running time

Density (likelihood) and gradient computation on a Intel(R) Core i7 CPU 1.9GHz 4GB RAM computer:

- **1.2s** using a pre-existing model,
- 23.0s taking a handmade Φ expression from the user,
- 26.4s without using C++ and using a pre-existing model.

$\Longrightarrow$ suggests the user to utilize pre-existing models.

Discussion and future work

- **CDN inference was difficult** because MPA implementation is complicated and no code was available.
- **MPA has been implemented** and CDN inference is easier.
- One now can fit CDN's to data.
- **Copula models** will be implemented in a future work.

References

[1] Huang, J.C. and Jojic, N. (2010). Maximum-likelihood learning of cumulative distribution functions on graphs. *13th International Conference on Artificial Intelligence and Statistics, AISTATS*.