High dimensional risk aggregation: a hierarchical approach with copulas

Philipp Arbenz

SCOR

Joint work with Christoph Hummel and Georg Mainik

RiskLab® Switzerland

ETH Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

SCOR

ESSEC, Paris, 24.5.2012

Risk aggregation: why?

Swiss Solvency Test (Solvency II is analogous):
one part of the solvency capital requirement (SCR) is

$$ES_{99\%}\left[\frac{Assets(1) - Liabilities(1)}{1+r} - (Assets(0) - Liabilities(0))\right],$$

where $Assets(t) - Liabilities(t)$ is the market consistent valuation of the available capital (= all assets minus all liabilities) at time t.

$Assets(1) - Liabilities(1)$ is random at time 0!

Risk aggregation: why?

One component for solvency capital requirements and risk management:

Calculate the distribution of *Liabilities*(1), which is the sum of all liabilities X_i:

$$Liabilities(1) = \sum_{i=1}^{d} X_i.$$

- X_i: value of liability i at time 1 (random at time 0)
- d: number of liabilities (usually huge!)

Risk aggregation: dependence matters

- Many risks are correlated
- Risks which are uncorrelated in “normal times” become dependent in the extremes.

  Examples:
  - 9/11 terrorist attacks
  - 2011 Tōhoku earthquake (Tsunami, Fukushima, etc.)

Risk aggregation: dependence matters

- Many risks are correlated
- Risks which are uncorrelated in "normal times" become dependent in the extremes.

  Examples:
  - 9/11 terrorist attacks
  - 2011 Tōhoku earthquake (Tsunami, Fukushima, etc.)

Dependence cannot be ingored!

Popular risk aggregation methodologies

- Variance - Covariance approaches
  - In high dimensions, number of correlation parameters $(= d(d-1)/2)$ becomes overwhelming
  - Conclusions are limited to statements on mean and (co-)variance

Popular risk aggregation methodologies

- Variance - Covariance approaches
  - In high dimensions, number of correlation parameters $(= d(d-1)/2)$ becomes overwhelming
  - Conclusions are limited to statements on mean and (co-)variance
- Risk factor models
  - Explicitly modelling risk factors can be difficult
  - Estimating risk factor sensitivities for all risks is challenging

Popular risk aggregation methodologies

- Variance - Covariance approaches
  - In high dimensions, number of correlation parameters $(= d(d-1)/2)$ becomes overwhelming
  - Conclusions are limited to statements on mean and (co-)variance
- Risk factor models
  - Explicitly modelling risk factors can be difficult
  - Estimating risk factor sensitivities for all risks is challenging
- Copula models
  - Can theoretically capture all aspects of dependence
  - Finding an adequate copula model is difficult - more details later

Copulas: definition

The joint cumulative distribution function (cdf) of $(X_1, \ldots, X_d)$ can be written as:

$$\mathbb{P}\left[X_1 \leq x_1, \ldots, X_d \leq x_d\right] = C\left(F_1(x_1), \ldots, F_d(x_d)\right),$$

where

- copula function $C : [0,1]^d \to [0,1]$
- marginal cdf's $F_i(x) = \mathbb{P}[X_i \leq x]$ $\qquad (F_i : \mathbb{R} \to [0,1])$

Copulas: definition

The joint cumulative distribution function (cdf) of $(X_1, \ldots, X_d)$ can be written as:

$$\mathbb{P}\left[X_1 \leq x_1, \ldots, X_d \leq x_d\right] = C\left(F_1(x_1), \ldots, F_d(x_d)\right),$$

where

- copula function $C : [0,1]^d \to [0,1]$
- marginal cdf's $F_i(x) = \mathbb{P}[X_i \leq x]$ $\qquad (F_i : \mathbb{R} \to [0,1])$

- C captures **all** aspects of dependence
- The F_i capture all aspects of the marginal distributions

Copula models

Setting up a copula model for the distribution of $(X_1, \ldots, X_d)$ is easy:

1. set a model for the F_i
2. set a model for C

Copula models

Setting up a copula model for the distribution of $(X_1, \ldots, X_d)$ is easy:

1. set a model for the F_i
2. set a model for C

There are many models for copulas:

- parametric
  - elliptic (Gaussian, t, ...)
  - Archimedean (Clayton, Gumbel, Frank, ...)
  - Vines
  - etc
- nonparametric
  - Bernstein copulas
  - Box copulas
  - Fourier copulas
  - etc

Problems with copulas in high dimensions

In high dimensions, most popular (parametric) copula classes are difficult to justify.

Possible issues are

- too symmetric dependence structure

Problems with copulas in high dimensions

In high dimensions, most popular (parametric) copula classes are difficult to justify.

Possible issues are

- too symmetric dependence structure
- difficult to calibrate
  - Not enough information/data
  - too many/too few parameters

Problems with copulas in high dimensions

In high dimensions, most popular (parametric) copula classes are difficult to justify.

Possible issues are

- too symmetric dependence structure
- difficult to calibrate
  - Not enough information/data
  - too many/too few parameters
- numerically slow simulation

Problems with copulas in high dimensions

In high dimensions, most popular (parametric) copula classes are difficult to justify.

Possible issues are

- too symmetric dependence structure
- difficult to calibrate
  - Not enough information/data
  - too many/too few parameters
- numerically slow simulation
- hard to justify (in front of management, regulators, rating agencies)

Problems with copulas in high dimensions

In high dimensions, most popular (parametric) copula classes are difficult to justify.

Possible issues are

- too symmetric dependence structure
- difficult to calibrate
  - Not enough information/data
  - too many/too few parameters
- numerically slow simulation
- hard to justify (in front of management, regulators, rating agencies)

Hierarchical risk aggregation circumvents these problems.

Hierarchical aggregation: example

We are interested in the aggregate

$$S = X_1 + X_2 + Y$$

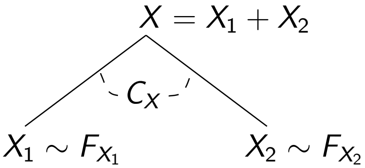
Suppose that modelling the copula of (X_1, X_2, Y) is not feasible.

Hierarchical aggregation: example

We are interested in the aggregate

$$S = X_1 + X_2 + Y$$

Suppose that modelling the copula of (X_1, X_2, Y) is not feasible.

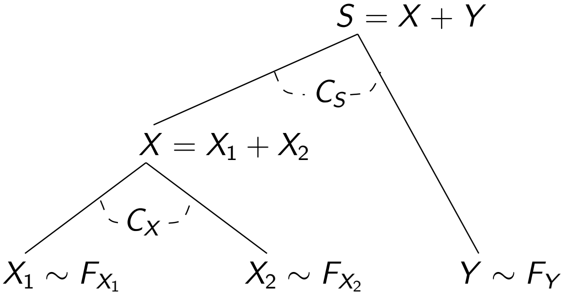


Hierarchical aggregation: example

We are interested in the aggregate

$$S = X_1 + X_2 + Y$$

Suppose that modelling the copula of (X_1, X_2, Y) is not feasible.



- F_{X_1}, F_{X_2}, F_Y: marginal cdf's of X_1, X_2, Y
- C_X: Copula of (X_1, X_2), C_S: Copula of (X, Y)

Existence & Uniqueness

Theorem: Suppose

- $X_1 \sim F_{X_1}, \quad X_2 \sim F_{X_2}, \quad Y \sim F_Y$ (margins)
- $(X_1, X_2) \sim C_X, \quad (X, Y) \sim C_S$ (copulas)
- $X = X_1 + X_2, \quad S = X + Y$ (aggregation steps)

Then

1. Distribution of (X_1, X_2, Y) exists, but is not unique
2. Distribution of (X_1, X_2) is unique. The same holds for (X, Y) and S

Existence & Uniqueness

Theorem: Suppose

- $X_1 \sim F_{X_1}, \quad X_2 \sim F_{X_2}, \quad Y \sim F_Y$ (margins)
- $(X_1, X_2) \sim C_X, \quad (X, Y) \sim C_S$ (copulas)
- $X = X_1 + X_2, \quad S = X + Y$ (aggregation steps)

Then

1. Distribution of (X_1, X_2, Y) exists, but is not unique
2. Distribution of (X_1, X_2) is unique. The same holds for (X, Y) and S

Consequences:

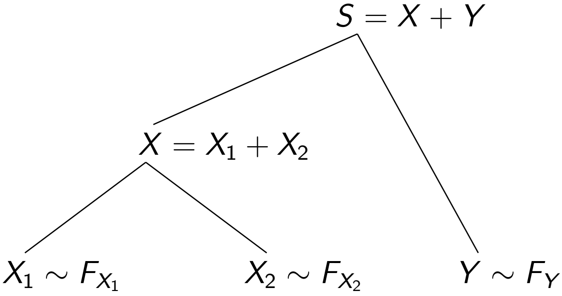
- If we are only interested in the aggregate S: no problem
- Classical capital allocation principles (e.g. Euler) do not work

Existence & Uniqueness

Theorem: Assume additionally that

- $(X_1, X_2) \perp Y | X$: Conditionally on X, (X_1, X_2) is independent of Y.

Then the distribution of (X_1, X_2, Y) is unique.



Modelling point of view

Classical approach: determine one trivariate copula describing the dependence structure of

$$(X_1, X_2, Y)$$

Modelling point of view

Classical approach: determine one trivariate copula describing the dependence structure of

$$(X_1, X_2, Y)$$

Hierarchical approach: determine 2 copulas C_X and C_S

$$(X_1, X_2) \sim C_X\left(F_{X_1}, F_{X_2}\right)$$
$$(X, Y) \sim C_S\left(F_X, F_Y\right)$$

Modelling point of view

Classical approach: determine one trivariate copula describing the dependence structure of

$$(X_1, X_2, Y)$$

Hierarchical approach: determine 2 copulas C_X and C_S

$$(X_1, X_2) \sim C_X\left(F_{X_1}, F_{X_2}\right)$$
$$(X, Y) \sim C_S\left(F_X, F_Y\right)$$

- These copulas are of lower dimension - "Divide & Conquer"

Reordering algorithm

Generating i.i.d. samples from the aggregation tree is not possible.

Instead: reordering algorithm for approximation.

Reordering algorithm

Generating i.i.d. samples from the aggregation tree is not possible.
Instead: reordering algorithm for approximation.

Illustration based on a simple problem:

$$(X, Y) \sim F_{X,Y} = C(F_X, F_Y).$$

1. Fix $n \in \mathbb{N}$
2. Simulate independently, for $i = 1, \ldots, n$:
   - $X_i \sim F_X$,
   - $Y_i \sim F_Y$,
   - $U_i = (U_i^1, U_i^2) \sim C$

Reordering algorithm

Generating i.i.d. samples from the aggregation tree is not possible.
Instead: reordering algorithm for approximation.

Illustration based on a simple problem:

$$(X, Y) \sim F_{X,Y} = C(F_X, F_Y).$$

1. Fix $n \in \mathbb{N}$
2. Simulate independently, for $i = 1, \ldots, n$:
   - $X_i \sim F_X$,
   - $Y_i \sim F_Y$,
   - $U_i = (U_i^1, U_i^2) \sim C$
3. Define an approximation of $F_{X,Y}$ by

$$F^n_{X,Y}(x, y) = C^n(F^n_X(x), F^n_Y(y))$$

The atoms of $F^n_{X,Y}$ are given by the samples of X and Y, reordered according to the observed joint ranks in the copula sample.

Reordering algorithm: Sampling margins and Copula

Let $n = 4$.

Sample i.i.d. $X_i \sim F_X$, $i = 1, 2, 3, 4$.

$X_i \sim F_X$		
sample		
3.1		
6.3		
1.4		
5.9		

Reordering algorithm: Sampling margins and Copula

Let $n = 4$.

Sample i.i.d. $X_i \sim F_X$, $i = 1, 2, 3, 4$.

$X_i \sim F_X$				
sample	rank			
3.1	2			
6.3	4			
1.4	1			
5.9	3			

Reordering algorithm: Sampling margins and Copula

Let $n = 4$.

Sample i.i.d. $X_i \sim F_X$, $i = 1, 2, 3, 4$.
Sample $Y_i \sim F_Y$ i.i.d., independent of the X_i.

$X_i \sim F_X$		$Y_i \sim F_Y$		
sample	rank	sample		
3.1	2	67.9		
6.3	4	22.8		
1.4	1	12.2		
5.9	3	43.7		

Reordering algorithm: Sampling margins and Copula

Let $n = 4$.

Sample i.i.d. $X_i \sim F_X$, $i = 1, 2, 3, 4$.
Sample $Y_i \sim F_Y$ i.i.d., independent of the X_i.

$X_i \sim F_X$		$Y_i \sim F_Y$		
sample	rank	sample	rank	
3.1	2	67.9	4	
6.3	4	22.8	2	
1.4	1	12.2	1	
5.9	3	43.7	3	

Reordering algorithm: Sampling margins and Copula

Let $n = 4$.

Sample i.i.d. $X_i \sim F_X$, $i = 1, 2, 3, 4$.
Sample $Y_i \sim F_Y$ i.i.d., independent of the X_i.
Sample $U_i \sim C$ i.i.d., $U_i \in [0, 1]^2$, independent of the X_i and Y_i.

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$
sample	rank	sample	rank	sample
3.1	2	67.9	4	(0.4,0.7)
6.3	4	22.8	2	(0.5,0.9)
1.4	1	12.2	1	(0.1,0.3)
5.9	3	43.7	3	(0.7,0.4)

Reordering algorithm: Sampling margins and Copula

Let $n = 4$.

Sample i.i.d. $X_i \sim F_X$, $i = 1, 2, 3, 4$.
Sample $Y_i \sim F_Y$ i.i.d., independent of the X_i.
Sample $U_i \sim C$ i.i.d., $U_i \in [0, 1]^2$, independent of the X_i and Y_i.

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$	
sample	rank	sample	rank	sample	rank
3.1	2	67.9	4	(0.4,0.7)	(2,3)
6.3	4	22.8	2	(0.5,0.9)	(3,4)
1.4	1	12.2	1	(0.1,0.3)	(1,1)
5.9	3	43.7	3	(0.7,0.4)	(4,2)

Reordering algorithm: Reordering

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$	
sample	rank	sample	rank	sample	rank
3.1	2	67.9	4	(0.4,0.7)	(2,3)
6.3	4	12.2	1	(0.5,0.9)	(3,4)
1.4	1	22.8	2	(0.1,0.3)	(1,1)
5.9	3	43.7	3	(0.7,0.4)	(4,2)

Samples of (X, Y):

Reordering algorithm: Reordering

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$	
sample	rank	sample	rank	sample	rank
3.1	2	67.9	4	(0.4,0.7) →	(2,3)
6.3	4	12.2	1	(0.5,0.9)	(3,4)
1.4	1	22.8	2	(0.1,0.3)	(1,1)
5.9	3	43.7	3	(0.7,0.4)	(4,2)

Samples of (X, Y):

Reordering algorithm: Reordering

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$	
sample	rank	sample	rank	sample	rank
3.1	2	67.9	4	(0.4,0.7) →	(2,3)
6.3	4	12.2	1	(0.5,0.9)	(3,4)
1.4	1	22.8	2	(0.1,0.3)	(1,1)
5.9	3	43.7	3	(0.7,0.4)	(4,2)

Samples of (X, Y):

Reordering algorithm: Reordering

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$	
sample	rank	sample	rank	sample	rank
3.1 ←	2	67.9	4	(0.4,0.7) →	(2,3)
6.3	4	12.2	1	(0.5,0.9)	(3,4)
1.4	1	22.8	2	(0.1,0.3)	(1,1)
5.9	3	43.7	3	(0.7,0.4)	(4,2)

Samples of (X, Y):

Reordering algorithm: Reordering

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$	
sample	rank	sample	rank	sample	rank
3.1	2	67.9	4	(0.4,0.7)	(2,3)
6.3	4	12.2	1	(0.5,0.9)	(3,4)
1.4	1	22.8	2	(0.1,0.3)	(1,1)
5.9	3	43.7	3	(0.7,0.4)	(4,2)

(3.1,

Samples of (X, Y):

Reordering algorithm: Reordering

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$	
sample	rank	sample	rank	sample	rank
3.1	2	67.9	4	(0.4,0.7)	(2,3)
6.3	4	12.2	1	(0.5,0.9)	(3,4)
1.4	1	22.8	2	(0.1,0.3)	(1,1)
5.9	3	43.7	3	(0.7,0.4)	(4,2)

(3.1,

Samples of (X, Y):

Reordering algorithm: Reordering

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$	
sample	rank	sample	rank	sample	rank
3.1	2	67.9	4	(0.4,0.7)	(2,3)
6.3	4	12.2	1	(0.5,0.9)	(3,4)
1.4	1	22.8	2	(0.1,0.3)	(1,1)
5.9	3	43.7	3	(0.7,0.4)	(4,2)

(3.1,

Samples of (X, Y):

Reordering algorithm: Reordering

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$	
sample	rank	sample	rank	sample	rank
3.1	2	67.9	4	(0.4,0.7)	(2,3)
6.3	4	12.2	1	(0.5,0.9)	(3,4)
1.4	1	22.8	2	(0.1,0.3)	(1,1)
5.9	3	43.7	3	(0.7,0.4)	(4,2)

$(3.1, 43.7)$

Samples of (X, Y):

Reordering algorithm: Reordering

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$	
sample	rank	sample	rank	sample	rank
3.1	2	67.9	4	(0.4,0.7)	(2,3)
6.3	4	12.2	1	(0.5,0.9)	(3,4)
1.4	1	22.8	2	(0.1,0.3)	(1,1)
5.9	3	43.7	3	(0.7,0.4)	(4,2)

Samples of (X, Y): $(3.1, 43.7)$

$(5.9, 67.9)$

Reordering algorithm: Reordering

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$	
sample	rank	sample	rank	sample	rank
3.1	2	67.9	4	(0.4,0.7)	(2,3)
6.3	4	12.2	1	(0.5,0.9)	(3,4)
1.4	1	22.8	2	(0.1,0.3)	(1,1)
5.9	3	43.7	3	(0.7,0.4)	(4,2)

Samples of (X, Y):

$(3.1, 43.7)$

$(5.9, 67.9)$

$(1.4, 12.2)$

Reordering algorithm: Reordering

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$	
sample	rank	sample	rank	sample	rank
3.1	2	67.9	4	(0.4,0.7)	(2,3)
6.3	4	12.2	1	(0.5,0.9)	(3,4)
1.4	1	22.8	2	(0.1,0.3)	(1,1)
5.9	3	43.7	3	(0.7,0.4)	(4,2)

Samples of (X, Y):

$(3.1, 43.7)$

$(5.9, 67.9)$

$(1.4, 12.2)$

$(6.3, 22.8)$

Reordering algorithm: Reordering

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$	
sample	rank	sample	rank	sample	rank
3.1	2	67.9	4	(0.4,0.7)	(2,3)
6.3	4	12.2	1	(0.5,0.9)	(3,4)
1.4	1	22.8	2	(0.1,0.3)	(1,1)
5.9	3	43.7	3	(0.7,0.4)	(4,2)

Samples of (X, Y): → Samples of $X + Y$:

$(3.1, 43.7) \longrightarrow 3.1 + 43.7 = 46.8$

$(5.9, 67.9) \longrightarrow 5.9 + 67.9 = 73.8$

$(1.4, 12.2) \longrightarrow 1.4 + 12.2 = 13.6$

$(6.3, 22.8) \longrightarrow 6.3 + 22.8 = 29.1$

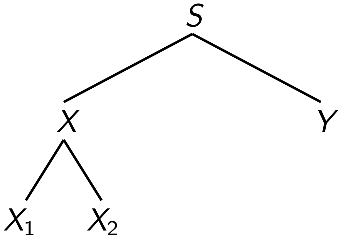
Reordering algorithm: Reordering

$X_i \sim F_X$		$Y_i \sim F_Y$		$U_i \sim C$	
sample	rank	sample	rank	sample	rank
3.1	2	67.9	4	(0.4,0.7)	(2,3)
6.3	4	12.2	1	(0.5,0.9)	(3,4)
1.4	1	22.8	2	(0.1,0.3)	(1,1)
5.9	3	43.7	3	(0.7,0.4)	(4,2)

Samples of (X, Y):
$$(3.1, 43.7)$$
$$(5.9, 67.9)$$
$$(1.4, 12.2)$$
$$(6.3, 22.8)$$

Samples of $X + Y$:
$$3.1 + 43.7 = 46.8$$
$$5.9 + 67.9 = 73.8$$
$$1.4 + 12.2 = 13.6$$
$$6.3 + 22.8 = 29.1$$

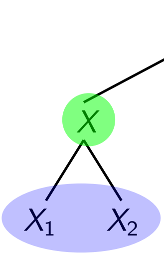
Sampling the aggregation tree



Dependence is described through 2 copulas:

$(X_1, X_2) \sim C_X \left(F_{X_1}, F_{X_2} \right)$ $\qquad$ $(X, Y) \sim C_S \left(F_X, F_Y \right)$

Sampling the aggregation tree



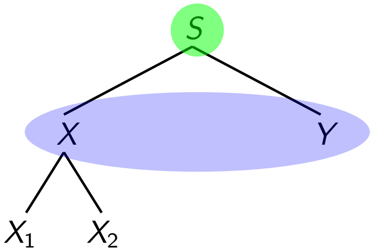
Dependence is described through 2 copulas:

$(X_1, X_2) \sim C_X(F_{X_1}, F_{X_2})$ $\qquad$ $(X, Y) \sim C_S(F_X, F_Y)$

Algorithm:

1. Reorder sample of X_1 and X_2 according to a sample of C_X. Sum components to get a sample of X.

Sampling the aggregation tree



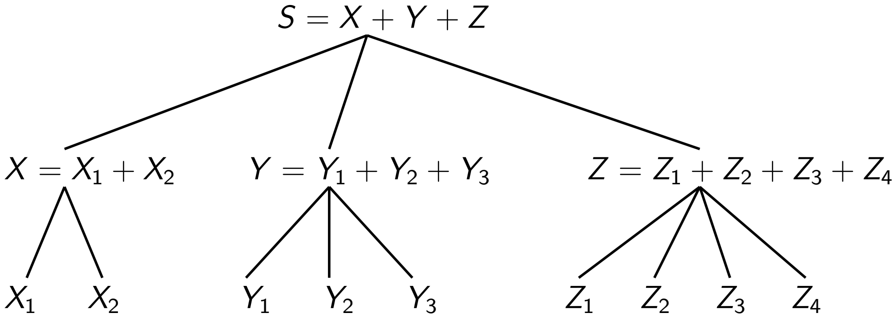
Dependence is described through 2 copulas:

$(X_1, X_2) \sim C_X\,(F_{X_1}, F_{X_2})$ $\qquad$ $(X, Y) \sim C_S\,(F_X, F_Y)$

Algorithm:

1. Reorder sample of X_1 and X_2 according to a sample of C_X.
   Sum components to get a sample of X.
2. Reorder sample of X and Y according to a sample of C_S.
   Sum components to get a sample of S.

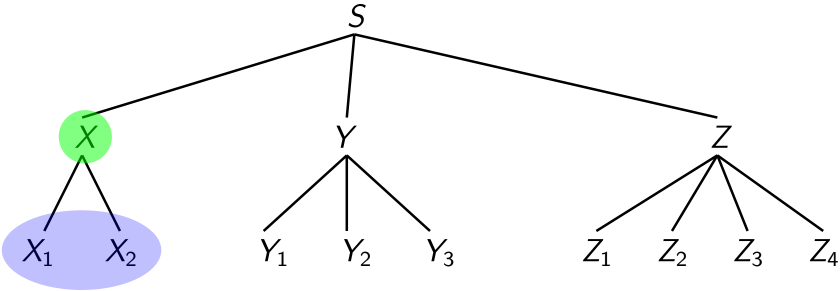
High dimensional example



Dependence is described through 4 copulas:

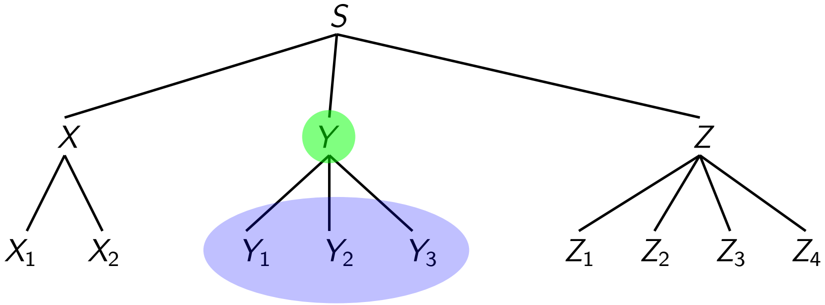
$$(X_1, X_2) \sim C_X\left(F_{X_1}, F_{X_2}\right)$$
$$(Y_1, Y_2, Y_3) \sim C_Y\left(F_{Y_1}, F_{Y_2}, F_{Y_3}\right)$$
$$(Z_1, Z_2, Z_3, Z_4) \sim C_Z\left(F_{Z_1}, F_{Z_2}, F_{Z_3}, F_{Z_4}\right)$$
$$(X, Y, Z) \sim C_S\left(F_X, F_Y, F_Z\right)$$

Aggregation example, sampling through reordering



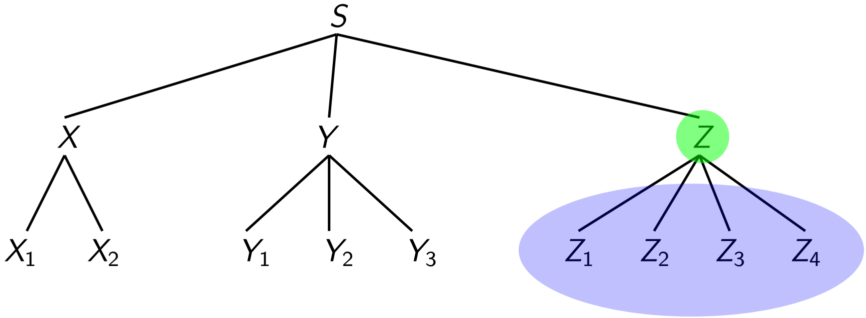
Reorder samples of X_1 and X_2 according to the copula C_X.
Calculate samples of X

Aggregation example, sampling through reordering



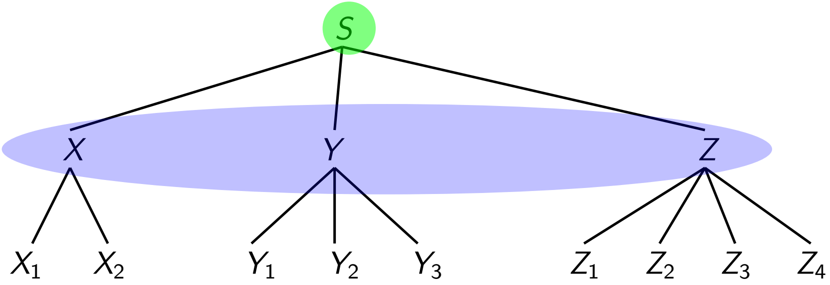
Reorder samples of Y_1, Y_2 and Y_3 according to the copula C_Y.
Calculate samples of Y

Aggregation example, sampling through reordering



Reorder samples of Z_1, Z_2, Z_3 and Z_4 according to the copula C_Z.
Calculate samples of Z

Aggregation example, sampling through reordering



Through the previous reorderings, we have samples of X, Y and Z!
Reorder those according to the copula C_T, in order to get samples of T.

Aggregation example, convergence

Theorem:
Suppose C_X and C_S are absolutely continuous with *bounded* densities. Then, the empirical cdf F_S^n of S converges to the true distribution F_S:

$$\lim_{n\to\infty}\sup_{x\in\mathbb{R}}\left|F_S^n(x)-F_S(x)\right| = 0 \qquad \mathbb{P}\text{-a.s.},$$

$$\sup_{x\in\mathbb{R}}\left|F_S^n(x)-F_S(x)\right| = O_{\mathbb{P}}(n^{-1/2}).$$

Aggregation example, convergence

Theorem:
Suppose C_X and C_S are absolutely continuous with *bounded* densities. Then, the empirical cdf F_S^n of S converges to the true distribution F_S:

$$\lim_{n\to\infty} \sup_{x\in\mathbb{R}} \left| F_S^n(x) - F_S(x) \right| = 0 \qquad \mathbb{P}\text{-a.s.},$$

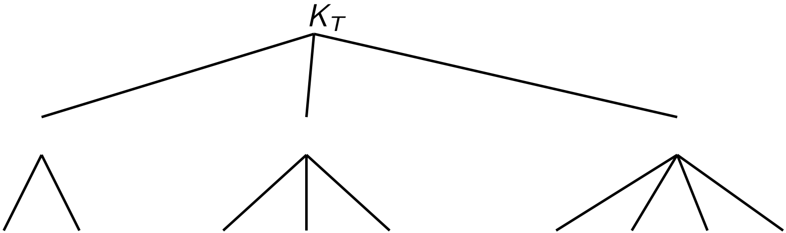
$$\sup_{x\in\mathbb{R}} \left| F_S^n(x) - F_S(x) \right| = O_{\mathbb{P}}(n^{-1/2}).$$

- Why only bounded densities? Underlying set classes **do not** satisfy Vapnik-Červonenkis (VC) property!
- For unbounded densities:
  - works for few examples (e.g. bivariate Clayton)
  - in general: open problem

Capital allocation

Capital allocation is easy: allocate hierarchically

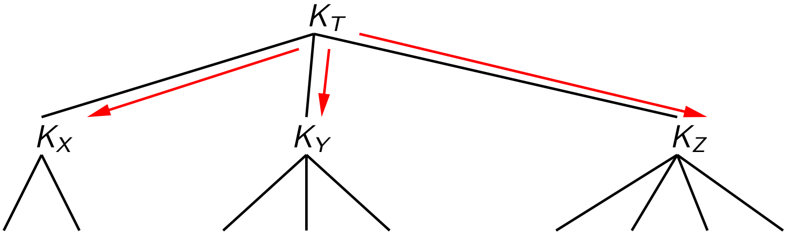
1. Risk capital: K_T



Capital allocation

Capital allocation is easy: allocate hierarchically

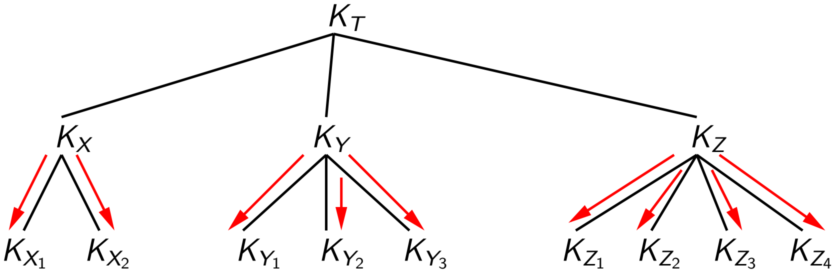
1. Risk capital: K_T
2. One has a sample of (X, Y, Z)!
   Allocate to X, Y, Z by splitting K_T



Capital allocation

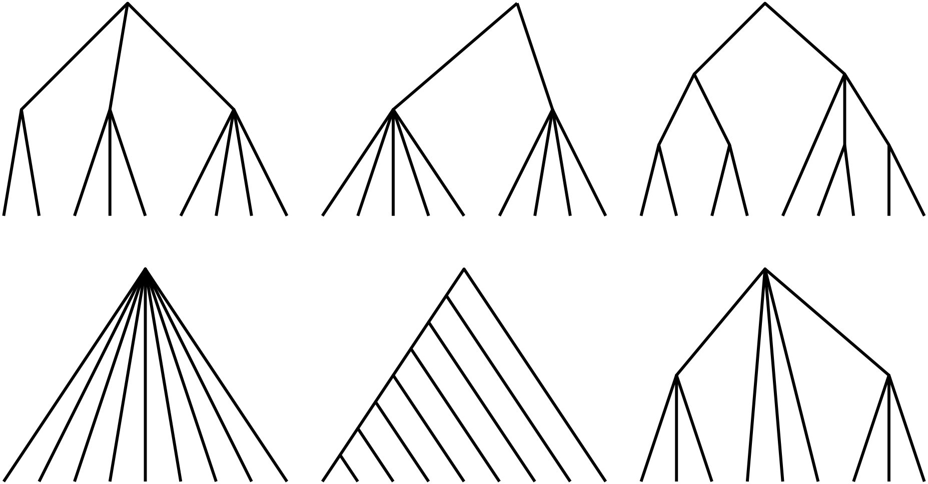
Capital allocation is easy: allocate hierarchically

1. Risk capital: K_T
2. One has a sample of (X, Y, Z)!
   Allocate to X, Y, Z by splitting K_T
3. One has a sample of (X_1, X_2)!
   Allocate to X_1 and X_2 by splitting K_X. Analogous for Y and Z



How to set the aggregation tree?

For 9 risks, there are 12'818'912 aggregation trees!



Estimation of the tree

Estimating the tree from data: not feasible. Model identification problems!

Estimation of the tree

Estimating the tree from data: not feasible. Model identification problems!

Heuristics:
Aggregate by risk types. Groupings are inherent due to

- line of business
- location
- maturity

Estimation of the tree

Estimating the tree from data: not feasible. Model identification problems!

Heuristics:
Aggregate by risk types. Groupings are inherent due to

- line of business
- location
- maturity

Dependence between risks gets weaker the farther they are apart

- Keep the number of aggregation levels low
- Strongest dependencies at the bottom
- Subaggregates with similar roles should be on the same level in the tree.

Estimation of copula parameters

As aggregation steps are low dimensional, the number of parameters is much lower than in a full model.

To calibrate copulas, use:

- Observations:
  - Maximum likelihood
  - Method of moments

Estimation of copula parameters

As aggregation steps are low dimensional, the number of parameters is much lower than in a full model.

To calibrate copulas, use:

- Observations:
  - Maximum likelihood
  - Method of moments
- Expert judgement
  - If the tree is cleverly chosen, aggregated variables for each aggregation step have intuitive meaning.

Estimation of copula parameters

As aggregation steps are low dimensional, the number of parameters is much lower than in a full model.

To calibrate copulas, use:

- Observations:
  - Maximum likelihood
  - Method of moments
- Expert judgement
  - If the tree is cleverly chosen, aggregated variables for each aggregation step have intuitive meaning.
- Regulatory guidelines

Or a combination of these sources of information.

Conclusion

- Very high dimensions are feasible

Conclusion

- Very high dimensions are feasible
- Flexible dependence structure
  - any type of copulas can be combined
  - any tree can be used
  - open statistical problem: how to "estimate" the tree?

Conclusion

- Very high dimensions are feasible
- Flexible dependence structure
  - any type of copulas can be combined
  - any tree can be used
  - open statistical problem: how to "estimate“ the tree?
- Simulation is easy

Conclusion

- Very high dimensions are feasible
- Flexible dependence structure
  - any type of copulas can be combined
  - any tree can be used
  - open statistical problem: how to "estimate" the tree?
- Simulation is easy
- Construction of a joint sample of the tree is possible (not shown here), but proof of convergence is still open

Conclusion

- Very high dimensions are feasible
- Flexible dependence structure
  - any type of copulas can be combined
  - any tree can be used
  - open statistical problem: how to "estimate" the tree?
- Simulation is easy
- Construction of a joint sample of the tree is possible (not shown here), but proof of convergence is still open
- Estimating the tree: heuristics

Conclusion

- Very high dimensions are feasible
- Flexible dependence structure
  - any type of copulas can be combined
  - any tree can be used
  - open statistical problem: how to "estimate" the tree?
- Simulation is easy
- Construction of a joint sample of the tree is possible (not shown here), but proof of convergence is still open
- Estimating the tree: heuristics
- Calibration is easy: only low dimensional problems

Conclusion

- Very high dimensions are feasible
- Flexible dependence structure
  - any type of copulas can be combined
  - any tree can be used
  - open statistical problem: how to "estimate" the tree?
- Simulation is easy
- Construction of a joint sample of the tree is possible (not shown here), but proof of convergence is still open
- Estimating the tree: heuristics
- Calibration is easy: only low dimensional problems
- Capital allocation is possible

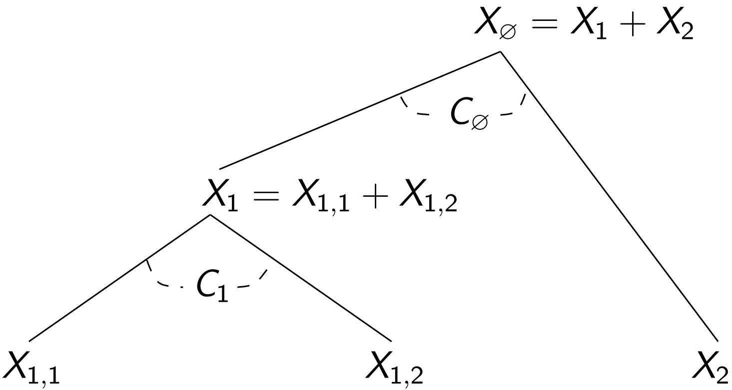
Conclusion

- Very high dimensions are feasible
- Flexible dependence structure
    - any type of copulas can be combined
    - any tree can be used
    - open statistical problem: how to "estimate" the tree?
- Simulation is easy
- Construction of a joint sample of the tree is possible (not shown here), but proof of convergence is still open
- Estimating the tree: heuristics
- Calibration is easy: only low dimensional problems
- Capital allocation is possible
- The model is graphically and mathematically intuitive: can be explained and defended in front of stakeholders

Thank you.

Appendix: Sampling the whole tree

Up to now: sampling described only for each aggregation step.
How to sample from the whole tree?
Idea: pull back permutations from top to bottom of the tree!



marginal samples: $\overset{X_{1,1}}{\begin{pmatrix}0\\0.1\\0.2\end{pmatrix}}$ $\overset{X_{1,2}}{\begin{pmatrix}0\\1\\2\end{pmatrix}}$ $\overset{X_2}{\begin{pmatrix}0\\10\\20\end{pmatrix}}$

reordering of $X_{1,1}$ and $X_{1,2}$: $\begin{matrix}X_{1,1} & X_{1,2}\end{matrix}$ $\begin{pmatrix}0.1 & 0\\0 & 2\\0.2 & 1\end{pmatrix} \to \sum \to \overset{X_1}{\begin{pmatrix}0.1\\2\\1.2\end{pmatrix}}$

reordering of X_1 and X_2: $\begin{matrix}X_1 & X_2\end{matrix}$ $\begin{pmatrix}2 & 10\\1.2 & 20\\0.1 & 0\end{pmatrix} \to \sum \to \overset{X_\varnothing}{\begin{pmatrix}12\\21.2\\0.1\end{pmatrix}}$

Apply to $X_{1,1}$ and $X_{1,2}$ the permutations which were applied to X_1. I.e., pull back permutations to leaf nodes to construct a joint sample:

$$\begin{matrix}X_{1,1} & X_{1,2} & X_1 & X_2 & X_\varnothing\end{matrix}$$
$$\begin{pmatrix}0 & 2 & 2 & 10 & 12\\0.2 & 1 & 1.2 & 20 & 21.2\\0.1 & 0 & 0.1 & 0 & 0.1\end{pmatrix}$$