

# Coupled Functions

© Copyright 2020-2025 Kenric P. Nelson

*Licensed under the Apache License, Version 2.0 (the “License”);  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at*

*<http://www.apache.org/licenses/LICENSE-2.0>*

*Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an “AS IS” BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.*

*Rev. 16 Kenric Nelson, October 2, 2025*

*Completed the Normalization of the Multivariate Coupled Stretched Exponential*

*Rev. 15 Kenric Nelson, February 15, 2024*

*Added Coupled Mean, CoupledCrossProb, and CoupledEntropyProb  
Corrected mistakes in the betaQToScale and scaleShapeToBeta functions*

*Rev. 14 Kenric Nelson, February 9, 2024*

*Revised Risk Bias to be defined as the relative risk aversion,*

$$r = \frac{\alpha \kappa}{1 + d \kappa}.$$

*Rev. 13 Kenric Nelson, December 27, 2022*

*Added qExponentialDistribution[ $\mu_-$ ,  $\beta_-$ ,  $q_-$ ] and qNormalDistribution[ $\mu_-$ ,  $\beta_-$ ,  $q_-$ ] to support comparisons with q-statistic formulations. These distribution functions translate  $\beta$  and  $q$  into  $\sigma$  and  $\kappa$  and then call CoupledExponentialDistribution and CoupledNormalDistribution respectively.*

*Rev. 12 Kenric Nelson, April 21, 2021*

*Revised CoupledExponential Function to an If statement based on whether  $(1 + \kappa x)$  is positive rather than on whether  $\kappa$  is positive. This corrects how negative values of  $x$  are handled and simplifies the function.*

*Rev. 11 Kenric Nelson*

*Added Coupled Box-Müller Method*

Rev. 10 Kenric Nelson

The compact-support domain of the `CoupledNormalDistribution` was producing a Power error message due  $1/\sqrt{0}$ . This was because `CoupledExponential` was defaulting to zero even for the reciprocal of zero. Updated `CoupledNormalDistribution` to be defined over the proper range for the compact-support domain. Updated `CoupledExponential` to evaluate to infinity for  $1/0$  situations. Also updated the `CoupledExponentialDistribution` functions.

The `CoupledMultivariateDistribution` function still needs attention.

Rev. 9

Add translators between the coupling ( $\kappa$ ), risk ( $r$ ), and Tsallis ( $q$ )

Rev. 8

a) Added clear and memory to each function. When the notebook is run it will clear the memory of that function. Each time the function is executed the result will be stored in memory, so that computation with the same inputs does not need to be repeated.

b) reordered coupling input to be consistent with `StudentsTDistribution` and related Wolfram functions

Rev. 7 Removed the parameter alpha from the definition of the coupled exponential and logarithm. This is preparation for experimenting on the proper two-parameter definitions for the functions.

Rev. 6 Added Weighted Generalized Mean

Rev. 5 Modified Coupled Product so that dimension used on outer exponent is  $d$  times `Length[xList]`

Rev. 4 Added Coupled Sine and Cosine Functions

Prototype initiated of Coupled Product and Coupled Sum with structure of *Mathematica* Product and Sum functions

## Coupled Exponential

The variable  $x$  is applied to a coupled exponential function.  $\kappa$  controls the degree of nonlinear statistical coupling, where  $\kappa = 0$  is linear domain of an exponential function.  $d$  and  $\alpha$  should match the dimension and power ( $x = y^\alpha$ ) of the variable  $x$ .

Inputs

$x$  - variable to which coupled exponential is applied

$\kappa$  - coupling parameter which modifies the coupled exponential function

$d$  - dimension, should be equal to the dimension of  $x$

```

In[1]:= ClearAll[CoupledExponential];
CoupledExponential[x_, κ_ : 0, d_ : 1] :=
CoupledExponential[x, κ, d] =
If[
  (1 + κ x) > 0,
  If[κ ≠ 0, (1 + κ x) $\frac{1+d κ}{κ}$ , Exp[x]],
  If[ $\frac{1+d κ}{κ}$  > 0, 0, ∞]
]

In[3]:= ClearAll[CoupledExponentialOld];
CoupledExponentialOld[x_, κ_ : 0, d_ : 1] :=
CoupledExponentialOld[x, κ, d] =
Which[
  κ > 0, (1 + κ x) $\frac{1+d κ}{κ}$ ,
  -1/d ≤ κ < 0, If[Simplify[1 + κ x] ≥ 0,
    (1 + κ x) $\frac{1+d κ}{κ}$ ,
    If[ $\frac{1+d κ}{κ}$  > 0, 0, ∞]
  ],
  κ == 0, Exp[x],
  True, Message[CoupledExponentialOld::nnarg, κ]
]

In[5]:= CoupledExponentialOld::nnarg = "Error: κ = `1/d` is not greater than -1.";

```

## Coupled Logarithm

Generalization of the logarithm function, which defines smooth transition to power functions.

Inputs

x - variable to which coupled exponential is applied

κ - coupling parameter which modifies the coupled exponential function

d - dimension, should be equal to the dimension of x

```

In[6]:= ClearAll[CoupledLogarithm];
CoupledLogarithm[x_, κ_ : 0, d_ : 1] :=
CoupledLogarithm[x, κ, d] =
If[x ≥ 0,
If[κ ≠ 0,
 $\frac{1}{\kappa} \left( x^{\frac{\kappa}{1+d\kappa}} - 1 \right),$ 
Log[x]
],
Undefined
]

```

## Coupled Gaussian - One Dimension

```

In[8]:= ClearAll[CoupledNormalDistribution];
CoupledNormalDistribution[μ_ : 0, σ_, κ_] :=
CoupledNormalDistribution[μ, σ, κ] =
Module[{x},
ProbabilityDistribution[
 $\frac{1}{\text{NormCG}[\sigma, \kappa]} \left( \text{CoupledExponential} \left[ \frac{(x - \mu)^2}{\sigma^2}, \kappa \right] \right)^{\frac{-1}{2}},$ 
If[κ ≥ 0,
{x, -∞, ∞},
{x,  $\mu - \sqrt{-\frac{\sigma^2}{\kappa}}$ ,  $\mu + \sqrt{-\frac{\sigma^2}{\kappa}}$ }
]
]
]

```

## Normalization of 1-D Coupled Gaussian

```

In[10]:= NormCG[σ_, κ_] :=  $\begin{cases} \sqrt{2\pi} \sigma & \kappa == 0 \\ \frac{\sqrt{\pi} \sigma \text{Gamma}\left[\frac{-1+\kappa}{2\kappa}\right]}{\sqrt{-\kappa} \text{Gamma}\left[1-\frac{1}{2\kappa}\right]} & \kappa < 0 \\ \frac{\sqrt{\pi} \sigma \text{Gamma}\left[\frac{1}{2\kappa}\right]}{\sqrt{\kappa} \text{Gamma}\left[\frac{1+\kappa}{2\kappa}\right]} & \text{True} \end{cases}$ 

```

## Coupled Exponential Distribution - One Dimension

The coupled exponential distribution is also known as the generalized Pareto distribution

```
In[11]:= ClearAll[CoupledExponentialDistribution];
CoupledExponentialDistribution[μ_ : 0, σ_, κ_] :=
CoupledExponentialDistribution[μ, σ, κ] =
Module[{x},
  ProbabilityDistribution[
     $\frac{1}{\sigma} \left( \text{CoupledExponential}\left[\frac{x - \mu}{\sigma}, \kappa, 1\right] \right)^{-1},$ 
    {x, μ, If[κ ≥ 0, ∞,  $\frac{-\sigma}{\kappa} + \mu$ ]}
  ]
]
```

## Multivariate Coupled Stretched Exponential Distribution

Status: Normalization has been updated but not tested

Currently the definition of multiple variables is not working properly

```

In[13]:= ClearAll[MultivariateCoupledDistribution];
MultivariateCoupledDistribution[μ_, Σ_, κ_, α_] :=
MultivariateCoupledDistribution[μ, Σ, κ, α] =
Module[{dimMean, dimCor, x},
(* Check that Σ is a positive
definite matrix with dimensions equal to length of μ *)
dimMean = Length[μ]; dimCor = Dimensions[Σ];
If[dimCor ≠ dimMean,
Message[MultivariateCoupledDistribution::argx, dimCor, dimMean]
];
(*If[Not@PositiveDefiniteMatrixQ[Σ],
Message[MultivariateCoupledDistribution::corr]
];*)
(*
If[α ≠ (1 || 2),
Message[MultivariateCoupledDistribution::alpha, α]
];
*)

(* Define the distribution *)
(* Evaluation of the input x to CoupledExponential is completed to insure
that the definition of CoupledExponential can stay one dimensional *)
x = Table[Symbol["$x" <> ToString@i], {i, dimMean}];
ProbabilityDistribution[
(
CoupledExponential[ ((x - μ).Inverse[Σ].(x - μ))α/2, κ, dimMean] )-1/α /
NormMultiCoupled[√Det[Σ], κ, α, dimMean],
(* Define the domain *)
###
] &@@
Which[α == 1, Table[{x[[i]], μ[[i]], ∞}, {i, dimMean}],
α == 2, Table[{x[[i]], -∞, ∞}, {i, dimMean}]
]
]
MultivariateCoupledDistribution::argx =
"Length of the first dimension of the matrix
`1` does not match the length of the mean `2`.";
MultivariateCoupledDistribution::corr =
"The correlation matrix Σ is not positive definite.";
(*
MultivariateCoupledDistribution::alpha="Alpha `1` is required to be 1 or 2";
*)

```

## Normalization of Multivariate Coupled Stretched Exponential

Solution is for the two-sided or radial distribution.

Example (Members Of The CEF)

Two broad groups of the CEF are when  $h(\mathbf{x}) = 1$  (the coupled stretched exponential distribution) and when the survival function is only  $\exp_{\alpha\kappa}^{-1}[\boldsymbol{\eta}(\boldsymbol{\theta}) \cdot \mathbf{T}(\mathbf{x})]$  (the coupled Weibull distribution). To simplify the expression, a radial variable is defined as  $r = |(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})|^{\frac{1}{2}} \geq 0$ .

### I. Coupled Stretched Exponential Distribution

PDF:

$$\begin{aligned}
 f_{\kappa}^{\text{Stretch}}(\mathbf{x}) &\equiv \frac{1}{Z_{\kappa}} (1 + \kappa r^{\alpha})^{-\frac{1+d\kappa}{\alpha\kappa}} \\
 &\equiv \frac{1}{Z_{\kappa}} \exp_{\alpha\kappa}^{-(1+\kappa)} \left( \frac{r^{\alpha}}{\alpha} \right) \\
 &\equiv \frac{1}{Z_{\kappa}} \exp_{\kappa}^{-\frac{1+\kappa}{\alpha}} (r^{\alpha}) \\
 Z_{\kappa} &= \frac{2\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2})} |\boldsymbol{\Sigma}|^{\frac{1}{2}} \begin{cases} \frac{1}{\alpha} \kappa^{-\frac{d}{\alpha}} B\left(\frac{d}{\alpha}, \frac{1}{\alpha\kappa}\right) & \kappa > 0 \\ \alpha^{(\frac{d}{\alpha}-1)} \Gamma\left(\frac{d}{\alpha}\right) & \kappa = 0 \\ \frac{1}{\alpha} (-\kappa)^{-\frac{d}{\alpha}} B\left(\frac{d}{\alpha}, 1 - \frac{1+d\kappa}{\alpha\kappa}\right) & -\frac{1}{d} < \kappa < 0. \end{cases}
 \end{aligned}$$

```

In[17]:= ClearAll[NormMultiCoupled];
NormMultiCoupled[sqrtDetSigma_, kappa_, alpha_, d_] :=
  NormMultiCoupled[sqrtDetSigma, kappa, alpha, d] =
    2 Pi^{d/2} / Gamma[d/2] sqrtDetSigma *
    Which[
      kappa > 0, 1/alpha kappa^{-d/alpha} Beta[d/alpha, 1/(alpha kappa)],
      kappa == 0, alpha^{d/alpha-1} Gamma[d/alpha],
      -1/d < kappa < 0, 1/alpha (-kappa)^{-d/alpha} Beta[d/alpha, 1 - (1+d kappa)/(alpha kappa)]
    ];

```

Save compact-support solution that was suggested by DeepSeek

$$-1/d < \kappa < 0, \quad \frac{1}{\alpha} (-\kappa)^{-d/\alpha} \text{Beta}\left[-\frac{1}{\alpha\kappa} + 1 - \frac{d}{\alpha}, \frac{d}{\alpha}\right]$$



Test function to confirm numerical integrate the multivariate stretched exponential

```

In[19]:= testmu = 1.1; testSigma = 2.3;

```

```
In[20]:= NIntegrate[PDF[
  MultivariateCoupledDistribution[testμ, testΣ, .24, 1.1],
  x],
  {x, 0, ∞}]
```

 **NIntegrate:** The integrand PDF [Null, x ] has evaluated to non -numerical values for all sampling points in the region with boundaries {{∞, 0.}}. 

```
Out[20]= NIntegrate[
  PDF[MultivariateCoupledDistribution[testμ, testΣ, 0.24, 1.1], x], {x, 0, ∞}]
```

## Coupled Probability

The Coupled Probability raises a distribution to the power

$1 + r$ , where  $r$  is the relative risk aversion and then renormalizes the distribution. The input

The coupling term  $M$  is the multiplicative coupling. It is

related to the source of coupling  $\kappa$  (small kappa) by the relationship

$$r = \frac{\alpha \kappa}{1 + d \kappa}$$

where  $\alpha$  is the power and  $d$  is the dimension of the random variable of the distribution

The function is currently written for continuous distributions,

though discrete distributions are also possible.  $f$  is required to be distribution, such that

$\text{PDF}[f, \{x, x_{\min}, x_{\max}\}]$  will produce a continuous distribution. The output is also a new distribution.



```

In[21]:= ClearAll[CoupledProbability];
CoupledProbability[dist_, r_, x_, xmin_ : (-∞), xmax_ : ∞] :=
CoupledProbability[dist, r, x, xmin, xmax] =
If[r == 0, PDF[dist, x],
FullSimplify[

$$\frac{(\text{PDF}[\text{dist}, x])^{1+r}}{\int_{x_{\min}}^{x_{\max}} (\text{PDF}[\text{dist}, y])^{1+r} dy}$$

]
]

ClearAll[NCoupledProbability];
NCoupledProbability[dist_, r_, x_, xmin_ : (-∞), xmax_ : ∞] :=
NCoupledProbability[dist, r, x, xmin, xmax] =
If[r == 0, PDF[dist, x],
(PDF[dist, x])1+r /
NIntegrate[(PDF[dist, y])1+r, {y, xmin, xmax}, Method → "AdaptiveMonteCarlo"]
]

```

## Biased Coupled Distribution

The Biased Coupled Distribution function provides an analytic solution for the Coupled Probability assuming the distribution is a member of the Coupled Exponential Family. The inputs are the parameters of the Coupled Exponential Family distribution and the coupling for the riskBias power. The output is a distribution from the Coupled Exponential Family with the coupling and scale modified by the riskBias power.

The first implementation focuses on multivariate coupled exponential distribution.

The modified coupling parameter for the output distribution is determined from the following relationship

$$\text{Solve}\left[\frac{1+d \kappa_{\text{Mod}}}{-\alpha \kappa_{\text{Mod}}} = \left(\frac{1+d \kappa_{\text{Dist}}}{-\alpha \kappa_{\text{Dist}}}\right) \left(1 - \frac{-\alpha \kappa_{\text{Bias}}}{1+d \kappa_{\text{Bias}}}\right), \kappa_{\text{Mod}}, \text{Reals}\right] // \text{FullSimplify}$$

$$\left\{\left\{\kappa_{\text{Mod}} \rightarrow \frac{\kappa_{\text{Dist}}+d \kappa_{\text{Bias}} \kappa_{\text{Dist}}}{1+\kappa_{\text{Bias}} (d+\alpha+d \alpha \kappa_{\text{Org}})}\right\}\right\}$$

The correlation matrix is multiplied by this terms divided by  $\kappa_{\text{Dist}}$  since  $\kappa_{\text{Dist}} \Sigma^{-1}$  is unchanged.

```

In[25]:= ClearAll[BiasedCoupledDistribution];
BiasedCoupledDistribution[μ_, Σ_, κDist_, α_, κBias_] :=
BiasedCoupledDistribution[μ, Σ, κDist, α, κBias] =
Module[{dim},
dim = Length[μ];
If[dim == 0, (* then the scalar is a one-dimensional distribution *)
If[α == 2,
CoupledNormalDistribution[
μ,  $\sqrt{\frac{1 + \kappa \text{Bias}}{1 + \kappa \text{Bias} (3 + 2 \kappa \text{Dist})}}$  Σ,  $\frac{\kappa \text{Dist} + \kappa \text{Bias} \kappa \text{Dist}}{1 + \kappa \text{Bias} (3 + 2 \kappa \text{Dist})}$  ],
CoupledExponentialDistribution[μ,  $\frac{1 + \kappa \text{Bias}}{1 + \kappa \text{Bias} (2 + \kappa \text{Dist})}$  Σ,  $\frac{\kappa \text{Dist} + \kappa \text{Bias} \kappa \text{Dist}}{1 + \kappa \text{Bias} (2 + \kappa \text{Dist})}$  ]
],
(* Else, assume dimension is greater than 2 *)
MultivariateCoupledDistribution[μ,
( $\frac{1 + \text{dim} \kappa \text{Bias}}{1 + \kappa \text{Bias} (\text{dim} + \alpha + \text{dim} \alpha \kappa \text{Dist})}$ ) $\frac{1}{\alpha}$  Σ,  $\frac{\kappa \text{Dist} + \text{dim} \kappa \text{Bias} \kappa \text{Dist}}{1 + \kappa \text{Bias} (\text{dim} + \alpha + \text{dim} \alpha \kappa \text{Dist})}$ , α]
]
]

```

## Coupled Entropy

Computes the coupled entropy using the coupled logarithm and coupled probability functions.

Currently limited to one-dimensional distributions

Updated Dec 17, 2020 to call the Coupled Cross-Entropy function

Inputs

p - probability distribution, limited to one-dimension

κ - coupling parameter

d - dimension of the random variable, default is 1

α - highest power of the constraints, default is 2 since this is associated with the Coupled Gaussian

δ - root of the coupled logarithm, default is 1; provides for two-parameter entropy experiments similar to c,d-Entropy

limits - domain of the variable; currently only one dimension is implemented

Output is the coupled entropy for a one-dimensional distribution

```

In[27]:= ClearAll[CoupledEntropy];
CoupledEntropy[dist_, κ_, d_ : 1, α_ : 2, δ_ : 1, limits_ : {-∞, ∞}] :=
  CoupledEntropy[dist, κ, d, α, δ, limits] =
    CoupledCrossEntropy[dist, dist, κ, α, d, δ, limits]

In[29]:= ClearAll[NCoupledEntropy];
NCoupledEntropy[dist_, κ_, d_ : 1, α_ : 2, δ_ : 1, limits_ : {-∞, ∞}] :=
  NCoupledEntropy[dist, κ, d, α, δ, limits] =
    NCoupledCrossEntropy[dist, dist, κ, d, α, δ, limits]

```

## Coupled Cross-Entropy

Computes the coupled cross-entropy using the coupled logarithm and coupled probability functions.  
Currently limited to one-dimensional distributions

Inputs

p - probability distribution, limited to one-dimension

κ - coupling parameter

d - dimension of the random variable, default is 1

α - highest power of the constraints, default is 2 since this is associated with the Coupled Gaussian

δ - root of the coupled logarithm, default is 1; provides for two-parameter entropy experiments similar to c,d-Entropy

limits - domain of the variable; currently only one dimension is implemented

Output is the coupled entropy for a one-dimensional distribution

```

In[31]:= ClearAll[CoupledCrossEntropy];
CoupledCrossEntropy[distP_, distQ_, κ_, d_ : 1, α_ : 2, δ_ : 1, limits_ : {-∞, ∞}] :=
  CoupledCrossEntropy[distP, distQ, κ, d, α, δ, limits] = Integrate[FullSimplify[
    CoupledProbability[distP,  $\frac{\alpha \kappa}{1 + d \kappa}$ , x]
     $\left( \frac{1}{\alpha} \text{CoupledLogarithm}[\text{PDF}[\text{distQ}, x]^{-\alpha}, \kappa, d] \right)^{\frac{1}{\delta}}$ 
  ], {x, limits[[1]], limits[[2]]}
]

```

```
In[33]:= ClearAll[NCoupledCrossEntropy];
NCoupledCrossEntropy[distP_, distQ_, κ_, d_:1, α_:2, δ_:1, limits_:{-∞, ∞}] :=
  (*NCoupledCrossEntropy[distP,distQ,κ,d,α,δ,limits]=*)
  NIntegrate[NCoupledProbability[distP,  $\frac{\alpha \kappa}{1 + d \kappa}$ , x]
     $\left(\frac{1}{\alpha} \text{CoupledLogarithm}[\text{PDF}[\text{distQ}, x]^{-\alpha}, \kappa, d]\right)^{\frac{1}{\delta}}$ , {x, limits[[1]], limits[[2]]},
    Method → "AdaptiveMonteCarlo"]
```

## Coupled Divergence

Computes the coupled divergence using its relationship with Coupled Entropy and Coupled Cross-Entropy

Currently limited to one-dimensional distributions

The definition of the Coupled Divergence is notionally:

Divergence = Cross-Entropy - Entropy

- CoupledProbability[distP, ...] (CoupledLogarithm[distQ, ...] - CoupledLogarithm[distP, ...])

This is equivalent to:

CoupledCrossEntropy[distP, distQ, ...] - CoupledEntropy[distP, ...]

note that for the first case the functional relationship for the generalization is the same as for  $\kappa = 0$ .

And there is an alternative form in which the division of the densities is preserved but the sum is generalized

For the second def

Inputs

p - probability distribution, limited to one-dimension

κ - coupling parameter

d - dimension of the random variable, default is 1

α - highest power of the constraints, default is 2 since this is associated with the Coupled Gaussian

δ - root of the coupled logarithm, default is 1; provides for two-parameter entropy experiments similar to c,d-Entropy

limits - domain of the variable; currently only one dimension is implemented

Output is the coupled entropy for a one-dimensional distribution

```

In[35]:= ClearAll[CoupledDivergence];
CoupledDivergence[distP_, distQ_, κ_, d_:1, α_:2, δ_:1, limits_:{-∞, ∞}] :=
  CoupledDivergence[distP, distQ, κ, d, α, δ, limits] =
    CoupledCrossEntropy[distP, distQ, κ, d, α, δ, limits] -
      CoupledEntropy[distP, κ, d, α, δ, limits]

In[37]:= ClearAll[NCoupledDivergence];
NCoupledDivergence[distP_, distQ_, κ_, d_:1, α_:2, δ_:1, limits_:{-∞, ∞}] :=
  NCoupledDivergence[distP, distQ, κ, d, α, δ, limits] =
    NCoupledCrossEntropy[distP, distQ, κ, d, α, δ, limits] -
      NCoupledEntropy[distP, κ, d, α, δ, limits]

```

## Coupled Product

The Coupled Product is a generalization of the product function, which raises each input to a power and after combining the inputs, takes the root. It is defined based on the sum of arguments of the coupled exponential:

$$\begin{aligned}
 \prod_{i=1}^N x_i &= \text{Exp}_{\alpha, \kappa, D} \left[ \sum_{i=1}^N \text{Log}_{\alpha, \kappa, d_i} [x_i] \right]; \quad D = \sum_{i=1}^N d_i \\
 &= \left( \sum_{i=1}^N x_i^{\frac{\alpha \kappa}{1+d_i \kappa}} - (N-1) \right)^{\frac{1+D \kappa}{\alpha \kappa}}; \quad D = \sum_{i=1}^N d_i
 \end{aligned}$$

### Inputs

xList - list of arguments to be combined by the coupled product

κ - coupling parameter

α - multiplicative term, default is 2 which is associated with coupled Gaussian distributions

d - dimension, currently defined as a constant, but planned to also except of list the same dimensions as xList; version 5 modified the dimensions of the outer exponent to be d times Length[xList]. Thus the inputs are assumed to be the same dimension and the output dimension is sum of the input dimensions.

Output - coupled product of xList

### Improvements Planned:

The dimension is defined as a constant. The dimension of each coupled logarithm should be controllable. The dimension of the coupled exponential should be the sum of the dimensions used for each coupled logarithm.

The CoupledProductPrototype is intended to allow the input of arguments to be combined match the structure of the built in *Mathematica* Product function. However; more extensive pattern matching is required to process the variety of inputs possible using this structure.

Old version: CoupledProductOld works but does not build upon the coupled exponential and coupled

logarithm functions. It does compute the output dimension as a sum of the input dimensions.

```
In[39]:= Clear[CoupledProduct];
CoupledProduct[xList_, κ_, α_ : 2, d_ : 1] :=
  CoupledProduct[xList, κ, α, d] =
    CoupledExponential[Total[CoupledLogarithm[#-α, κ, d] & /@ xList] $\frac{-1}{\alpha}$ ,
      κ, d Length[xList]];
```

## Coupled Sum

For two inputs the coupled sum is  $x + y + (-\alpha \kappa) x y$

For more than two inputs, the solution is determined using recursion. The last element of the list is extracted as one of the variables, and the other variable calls the coupled sum without the last element of the list.

The input is a list of variables to be operated on, the nonlinear coupling  $\kappa$ , and the power  $\alpha$  affecting the variable  $x^\alpha$ . The operator currently assumes each variable in the list is one-dimensional.

Improvement Planned

A simpler specification of the function uses the fold function. Use of alpha not equal to 1 still in evaluation. Earlier plan no longer necessary:

CoupledSumPrototype will define the coupled sum in terms of the product of arguments of the coupled logarithm. The input is intended to follow the structure of the *Mathematica* Sum function, but requires additional pattern matching of input types to be fully functional.

```
In[41]:=
```

```
In[42]:= Clear[CoupledSum];
CoupledSum[xList_, κ_, α_] :=
  CoupledSum[xList, κ, α] =
    If[Length[xList] == 2,
      First[xList] + Last[xList] + (-α κ) First[xList] × Last[xList],
      (*Else when xList > 2*)
      CoupledSum[Drop[xList, -1], κ, α] +
      Last[xList] +
      (-α κ) CoupledSum[Drop[xList, -1], κ, α] × Last[xList]
    ] /; Length[xList] > 1 (* only execute if length > 1 *)
```

```
CoupledSumPrototype[list_, κ_, α_ : 1] :=
  Fold[(#1α + #2α + κ #1α #2α) $\frac{1}{\alpha}$  &, list];
```

## Coupled Sine and Cosine

```
In[45]:= ClearAll[CoupledSin];
CoupledSin[x_, κ_, α_ : 1, d_ : 1] :=
CoupledSin[x, κ, α, d] =
  (CoupledExponential[i x, κ, d]1/α -
    CoupledExponential[-i x, κ, d]1/α) / (2 i)
```

```
In[47]:= ClearAll[CoupledCos];
CoupledCos[x_, κ_, α_ : 1, d_ : 1] :=
CoupledCos[x, κ, α, d] =
  (CoupledExponential[i x, κ, d]-1/α +
    CoupledExponential[-i x, κ, α, d]-1/α) / 2
```

## Weighted Generalized Mean

Computes the weighted generalized mean of a list. For a 2-D list the weighted generalized mean of each column is computed. The weights are normalized, thus weights equal to one correspond to the generalized mean

$$Y = \left( \sum_{i=1}^N \frac{W_i}{\sum_{i=1}^N W_i} X_i^p \right)^{\frac{1}{p}}$$

Inputs:

X - a one or two dimensional list of variables

W - scalar, vector, or matrix of weights which modify the contribution of each input  $X_i$

p - scalar, power for the generalized mean

p = -1 is the harmonic mean

p = 0 is the geometric mean

p = 1 is the arithmetic mean

p = 2 is the root mean square

Outputs:

Y - scalar or vector of weighted generalized mean; for 2-D input mean of the first dimension is computed

```
In[49]:= ClearAll[WeightedGeneralizedMean];
WeightedGeneralizedMean[X_, p_, W_ : 1] :=
WeightedGeneralizedMean[X, p, W] =
  Module[{n, m, i, j, normW, sumW, XArrayDepth, WArrayDepth, Y},

    XArrayDepth = ArrayDepth[X];
```

```

Which[
  XArrayDepth == 1, {n} = Dimensions[X],
  XArrayDepth == 2, {n, m} = Dimensions[X],
  XArrayDepth > 2, WeightedGeneralizedMean::argx =
    "Warning: Inputs X expected to have ArrayDepth of 2"
];

If[Length[p] > 0, WeightedGeneralizedMean::argx =
  "Warning: Input p expected to be a scalar"
];
Assert[True];
(* Check dimensions of W and initialize normW
  expand scalar to matrix size of X
  expand column vector into n rows
  or set message warning about mismatch in size*)
WArrayDepth = ArrayDepth[W];
If[XArrayDepth == 1,
  Which[WArrayDepth == 0, normW = Table[W, n],
    WArrayDepth == 1, normW = W,
    WArrayDepth > 1, WeightedGeneralizedMean::argx =
      "Warning: Input W expected to have same or less depth than X"
  ],
  (*Else if XArrayDepth == 2 *)
  Which[WArrayDepth == 0, normW = Table[W, {n}, {m}],
    WArrayDepth == 1, normW = Table[W, m] // Transpose,
    WArrayDepth == 2, normW = W,
    WArrayDepth > 2, WeightedGeneralizedMean::argx =
      "Warning: Input W expected to have ArrayDepth of 1 or 2"
  ];
];

If[Dimensions[normW] != Dimensions[X], WeightedGeneralizedMean::argx =
  "Dimensions of X and resized W do not match"];

If[XArrayDepth == 1, Module[{},
  Assert[True];
  sumW =  $\sum_{i=1}^n \text{normW}_{[[i]]}$ ;
  normW =  $\frac{\text{normW}}{\text{sumW}}$ ;
  Quiet[If[p != 0,
    Check[Y =  $\left(\sum_{i=1}^n \text{normW}_{[[i]]} X_{[[i]]}^p\right)^{\frac{1}{p}}$  // N,

```



```

      Y = If[p > 0, Max[X], Min[X]]
    ],
    Y =  $\prod_{i=1}^n X_{[[i]]}^{\text{normW}_{[[i]]}}$  // N
  ]]
],
Module[{ },
  Y = Table[0, m];

  (* Loop column and rows to compute weighted generalized mean *)
  For[j = 1, j ≤ m, j++,
    sumW =  $\sum_{i=1}^n \text{normW}_{[[i,j]]}$ ;
    normW[[All,j]] =  $\frac{\text{normW}_{[[All,j]]}}{\text{sumW}}$ ;
    Quiet[If[p ≠ 0,
      Check[Y[[j]] =  $\left( \sum_{i=1}^n \text{normW}_{[[i,j]]} X_{[[i,j]]}^p \right)^{\frac{1}{p}}$  // N,
      Y[[j]] = If[p > 0, Max[X[[All,j]]], Min[X[[All,j]]]]
    ],
    Y[[j]] =  $\prod_{i=1}^n X_{[[i,j]]}^{\text{normW}_{[[i,j]]}}$  // N
  ]]
]
];
(Print[sumW];*)
Assert[True];
N[Y, 5] (* Set output of module *)
(*{normW,XArrayDepth,WArrayDepth} *) (* Monitor Variables *)
];

```

## Coupled Probability Metrics

In[50]:= (\* The CoupledMean takes as inputs:

- 1) discrete variable - X,
- 2) discrete probability weights - P,
- 3) relative risk aversion - r,

and outputs the Coupled Mean

$$\text{CoupledMean}[X, r, P] = \left( \sum_{i=1}^N \frac{p_i^{1+r}}{\sum_{j=1}^N p_j^{1+r}} x_i^r \right)^{\frac{1}{r}}.$$

This is the weighted generalized mean with weights  $\frac{p_i^{1+r}}{\sum_{j=1}^N p_j^{1+r}}$ .

The CoupledCrossProbs[P,Q,r] is the Coupled Mean applied to probabilities Q.

CoupledCrossProb[P,Q,r]=CoupledMean[Q<sup>-1</sup>,r,P]<sup>-1</sup>. This function is also equal to CoupledExponential[-CoupledCrossEntropy[P,Q,κ,d,α],κ,d,α]

The CoupledDivergenceProbs[P,Q,r]=CoupledMean[P\*Q<sup>-1</sup>,r,P]<sup>-1</sup>

The CoupledEntropyProbs[P,r]= CoupledMean[P<sup>-1</sup>,r,P]<sup>-1</sup>

The definitions of the Coupled Divergence will need to be reexamined so that there is consistency with the probability expression. Since P\*Q<sup>-1</sup> is being used the divergence will also need to use this instead of CoupledLog[P]- CoupledLog[Q]. Also it is worthwhile to experiment with the two forms of divergence to see if either has more merit.

\*)

CoupledMean[X\_, r\_, P\_] :=

WeightedGeneralizedMean[X, r, P<sup>1+r</sup>/Table[Total[P], Dimensions[P][[1]]];

CoupledCrossProb[P\_, Q\_, r\_] :=

CoupledMean[Q<sup>-1</sup>, r, P]<sup>-1</sup>;

CoupledEntropyProb[P\_, r\_] := CoupledCrossProb[P, P, r];

(\* Add CoupledDivergenceProb as a function of CoupledCrossProb and CoupledEntropyProb once analysis on relationship is completed \*)

## Variable transformations

The relative risk aversion is  $r = \frac{\alpha \kappa}{1+d \kappa}$

risk and coupling Domain Dual

In[53]:= riskDomainDual[r\_, α\_ : 2, d\_ : 1] :=

riskDomainDual[r, α, d] =

$$\frac{-r \alpha}{\alpha - d r};$$

```
In[54]:= couplingDomainDual[κ_, α_ : 2, d_ : 1] := couplingDomainDual[κ, α, d] =
      - κ
      1 + d κ ;
```

Coupling to Relative Risk Aversion and Relative Risk Aversion to Coupling

```
In[55]:= couplingToRisk[κ_, α_ : 2, d_ : 1] := couplingToRisk[κ, α, d] =
      α κ
      1 + d κ ;
```

```
In[56]:= riskToCoupling[r_, α_ : 2, d_ : 1] := riskToCoupling[r, α, d] =
      r
      α - d r ;
```

```
In[57]:= qDomainDual[q_, α_, d_] :=
      qDomainDual[q, α, d] =
      d - d q + 2 α - q α
      d - d q + α ;
```

qToCoupling and couplingToq

The q-Gaussian is defined as  $\frac{\sqrt{\beta}}{C_q} e_q(-\beta_q x^2)$ ;  $e_q(y) = (1 + (1 - q)y)^{\frac{1}{1-q}}$ ;  $C_q = \sqrt{\frac{\pi}{q-1}} \frac{\Gamma[\frac{3-q}{2q-2}]}{\Gamma[\frac{1}{q-1}]}$

```
In[58]:= couplingToq[κ_, α_ : 2, d_ : 1] := couplingToq[κ, α, d] =
      1 + couplingToRisk[κ, α, d];
```

```
In[59]:= qToCoupling[q_, α_ : 2, d_ : 1] := qToCoupling[q, α, d] =
      - (1 - q)
      α + d (1 - q)
```

Beta  $\Leftrightarrow$  Scale is determined from  $-\beta(1 - q) = \frac{\kappa}{\sigma^\alpha}$ ;

$$-\beta\left(1 - 1 - \frac{\alpha \kappa}{1 + d \kappa}\right) = \frac{\kappa}{\sigma^\alpha}$$

$$\frac{\alpha \beta}{1 + d \kappa} = \frac{1}{\sigma^\alpha}$$

$$\beta = \frac{1 + d \kappa}{\alpha \sigma^\alpha};$$

$$\sigma^\alpha = \frac{1 + d \kappa}{\alpha \beta} = \frac{1 + d \frac{(-1 + q)}{\alpha + d(1 - q)}}{\alpha \beta}$$

```
In[60]:= 1 + d (-1 + q)
      α + d (1 - q) // FullSimplify
      α β
```

```
Out[60]=
      1
      (d - d q + α) β
```

```

In[61]:= betaQToScale[beta_, q_, α_ : 2, d_ : 1] :=  $\left( \frac{1}{(d (1 - q) + \alpha) \text{beta}} \right)^{\frac{1}{\alpha}};$ 

scaleShapeToBeta[σ_, κ_, α_ : 2, d_ : 1] :=  $\frac{1 + d \kappa}{\alpha \sigma^\alpha};$ 

Cq[q_] :=  $\sqrt{\frac{\pi}{(q - 1)}} \frac{\text{Gamma}\left[\frac{3-q}{2 q-2}\right]}{\text{Gamma}\left[\frac{1}{q-1}\right]};$ 

betaQToP0[β_, q_] :=  $\frac{\sqrt{\beta}}{\text{Cq}[q]};$ 

scaleShapeToP0[σ_, κ_] :=  $\frac{1}{\text{NormCG}[\sigma, \kappa]};$ 

P0QToBeta[P0_, q_, α_ : 2] := (P0 Cq[q])α;

P0QToScale[P0_, q_, α_ : 2] := betaQToScale[P0QToBeta[P0, q], α];

```

## Coupled Box-Müller Method

The coupled Box-Muller algorithm than has the following procedure.

- 1) Draw two uniform random variables U1 and U2 over the range 0 to 1.
- 2) Apply the generalized transformation  $(Z_1, Z_2) = T(U_1, U_2)$  where T is

$$Z_1 = \sqrt{\ln_\kappa(U_1^{-2})} \cos(2 \pi U_2)$$

$$Z_2 = \sqrt{\ln_\kappa(U_2^{-2})} \sin(2 \pi U_1).$$

In this case the dimension of the logarithm is zero and thus the variable U does not need to be raised to a power.

- 3)  $(Z_1, Z_2)$  forms a two-dimensional coupled Gaussian distribution with  $\mu_1 = \mu_2 = 0$  and  $\Sigma^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ .

For a set of independent random variables only one of the two variates should be selected, since although uncorrelated (no cross terms in the correlation matrix) the variates are dependent since the joint distribution cannot be factored.

- 4) The location and scale for the coupled Gaussian variates are added to Z by  $X = \mu + \sigma Z$ .
- 5)  $n$  is the number of samples to generate

In[68]:=

```

Clear[CoupledVariate];
CoupledVariate[μ_ : 0, σ_ : 1, κ_, n_ : 1] := Module[
  {UniformVariates = RandomReal[{0, 1}, 2 n], CoupledNormalVariates},
  CoupledNormalVariates =
    If[n == 1,
      Sqrt[CoupledLogarithm[UniformVariates[[1]]^-2, κ, 0]] Cos[2 π UniformVariates[[1]]],
      Table[
        Sqrt[CoupledLogarithm[UniformVariates[[i]]^-2, κ, 0]] Cos[2 π UniformVariates[[i]]],
        {i, n}
      ]
    ];
  μ + σ CoupledNormalVariates
];

```

## q-Statistics Distributions

```

In[70]:= qExponentialDistribution[μ_, β_, q_] := qExponentialDistribution[μ, β, q] =
  CoupledExponentialDistribution[μ, betaQToScale[β, q, 1], qToCoupling[q, 1, 1]];

In[71]:= qNormalDistribution[μ_, β_, q_] := qNormalDistribution[μ, β, q] =
  CoupledNormalDistribution[μ, betaQToScale[β, q, 2], qToCoupling[q, 2, 1]];

```