EnsAnam

Ensemble Anamorphosis Transformation

User's guide

Jean-Michel Brankart

`http://pp.ige-grenoble.fr/pageperso/brankarj/`

Institut des Géosciences de l'Environnement
Université Grenoble Alpes, CNRS, France

The purpose of EnsAnam is to provide tools to apply anamorphosis transformation to ensemble simulations. The objective is to transform the marginal distribution of all variables to the same distribution defined by the user (usually a normalized Gaussian distribution).

The tools are provided as a library of modules, which can be easily plugged in any existing software. This library includes:

- the computation of quantiles of the input ensemble (defining the transformation),
- the application of the transformation to any state vector,
- the transformation of observations.

1 Description of the method

The purpose of anamorphosis transformation is to apply a nonlinear transformation to the simulated variables, so that all marginal distributions become identical (usually a normalized Gaussian distribution). This is done by computing the transformation A_i associated to each variable x_i of the vector $\mathbf{x}$, so that $z_i = A_i(x_i)$ has the required marginal distribution, usually $\mathcal{N}(0,1)$. By combining these univariate transformations, we can write the transformed vector: $\mathbf{z} = A(\mathbf{x})$.

Our basic assumptions to compute the transformation A are that: (i) the probability distribution of $\mathbf{x}$ is described by an ensemble of moderate size, so that the transformation A can only be approximately identified, and (ii) the size of the vector $\mathbf{x}$ can be very large so that the practical algorithm (to compute and apply A and A^{-1}) must contain as few operations as possible.

Potential applications of this method are numerous, but the target application for which these modules have been developed is the transformation of the ensemble observational update so that the prior marginal distributions become Gaussian. The ensemble observational update is based on the Bayes theorem:

$$p^a(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}^o) = p^f(\mathbf{x}) \;\; p\left[\mathbf{y}^o|\mathcal{H}(\mathbf{x})\right] \tag{1}$$

where $p^f(\mathbf{x})$ is the prior probability distribution for the state $\mathbf{x}$ of the system, described by the prior ensemble, and $p^a(\mathbf{x})$ is the posterior probability distribution after the conditioning of $p^f(\mathbf{x})$ on observations $\mathbf{y}^o$. In the above formula, $\mathcal{H}$ is the observation operator, computing the observation equivalent from the state vector, and $p\left[\mathbf{y}^o|\mathcal{H}(\mathbf{x})\right]$ is the probability distribution for observations given the state $\mathbf{x}$ of the system.

With anamorphosis, the observational update can be performed using the transformed state vector $\mathbf{z}$ rather than the original state vector $\mathbf{x}$, and the result can be transformed back to the original variable using the inverse transformation A^{-1}. In the general case, the transformation of $\mathbf{x}$ to $\mathbf{z}$ also requires including the backward transformation A^{-1} in the observation operator[1] to compute the observation equivalent: $\mathbf{y} = \mathcal{H}\left[A^{-1}(\mathbf{z})\right]$. However, anamorphosis transformation of observations can sometimes be a better option (as explained in section 1.5).

1.1 Computation of the transformation

Let $F(x)$ be the cumulative distribution function (cdf) corresponding to the marginal probability distribution of a variable x of the state vector, and $G(z)$ be the cdf of the target distribution (usually a Gaussian distribution). Then, the forward and backward anamorphosis transformation, transforming x to z and z to x are given by:

$$z = G^{-1}\left[F(x)\right] \quad \text{and} \quad x = F^{-1}\left[G(z)\right] \tag{2}$$

The whole problem thus reduces to estimating $F(x)$ from the available ensemble (i.e. from a sample of the probability distribution).

A simple and numerically efficient solution to this problem (see Brankart et al., 2012, for more details) is to describe $F(x)$ by a set of quantiles $\tilde{x}_k$ of the ensemble, corresponding to the ranks r_k, $k = 1, \ldots, q$ [i.e. such that $F(\tilde{x}_k) = r_k$], and by linear interpolation between the quantiles. The transformation functions (corresponding to every variable x of the state vector) are this completely described by the quantiles of the ensemble, which can be obtained from the module **anaqua**.

[1] This can be done without any problem if the observational update is general enough to deal with nonlinear observation operators, especially if A^{-1} is not too expensive as compared to $\mathcal{H}$. However, in many simple situations (for instance if $\mathcal{H}$ is linear or if the observational update can only deal with a linear $\mathcal{H}$), it can be necessary to apply anamorphosis transformation to $\mathbf{y} = \mathcal{H}(\mathbf{x})$ and transform $p(\mathbf{y}^o|\mathbf{y})$ accordingly.

1.2 Application of the transformation

The transformation is then piecewise linear and work by remapping the quantiles $\tilde{x}_k$ of the ensemble on the corresponding quantiles $\tilde{z}_k$ of the target distribution:

$$A(x) = \tilde{z}_k + \frac{\tilde{z}_{k+1} - \tilde{z}_k}{\tilde{x}_{k+1} - \tilde{x}_k}(x - \tilde{x}_k) \quad \text{for} \quad x \in [\tilde{x}_k, \tilde{x}_{k+1}] \tag{3}$$

$$A^{-1}(z) = \tilde{x}_k + \frac{\tilde{x}_{k+1} - \tilde{x}_k}{\tilde{z}_{k+1} - \tilde{z}_k}(z - \tilde{z}_k) \quad \text{for} \quad z \in [\tilde{z}_k, \tilde{z}_{k+1}] \tag{4}$$

This transformation is monotonous and bijective between the intervals $[\tilde{x}_1, \tilde{x}_q]$ and $[\tilde{z}_1, \tilde{z}_q]$, providing that the quantiles are all distinct. See section 1.3 for a generalization to discrete events with finite probability (leading to non-distinct quantiles) and section 1.4 for a generalization to extreme events (outside the interval $[\tilde{x}_1, \tilde{x}_q]$). The direct consequence of these properties is that anamorphosis transformation preserves the rank of the ensemble members and thus the rank correlation between variables (see Brankart et al., 2012, for more details about the effect of the transformation on correlations).

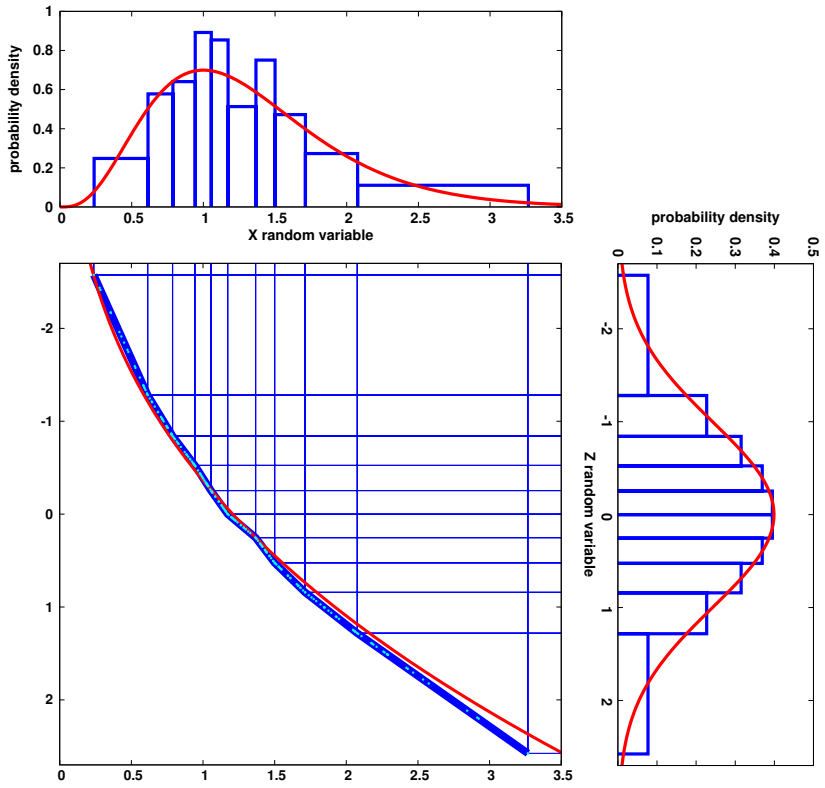

Figure 1: Approximate piecewise linear anamorphosis transformation (thick blue curve), remapping the deciles $\tilde{x}_k$ of a 200-member random sample of the Gamma distribution $\Gamma(k, \theta)$ (top histogram) on the Gaussian deciles $\tilde{z}_k$ (left histogram), as compared to the exact transformation (in red) transforming the exact $\Gamma(k, \theta)$ (red curve superposed to the top histogram) into $\mathcal{N}(0, 1)$ (red curve superposed to the left histogram).

Figure 1 illustrates the behaviour of the algorithm by the transformation of a gamma distribution into a Gaussian distribution, using a 200-member ensemble. The application of the piecewise linear transformation (in blue in the figure) can be performed using the module **anatra** (using the ensemble quantiles provided by **anaqua**).

1.3 Discrete events

In many practical applications, there can be problems in which a finite probability concentrates on some critical value x_c of the state variable. In this case the cdf $F(x)$ is discontinuous and the standard anamorphosis transformation described by Eq. (3) does not apply.

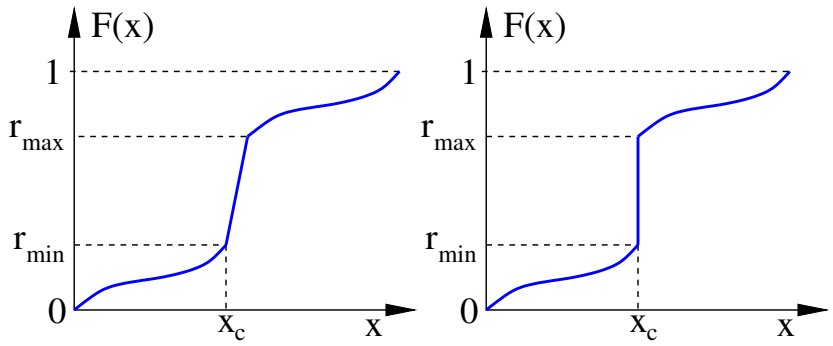

Figure 2: As long as there is a slope in the cdf (left panel), we know which value of the rank $r = F(x)$ corresponds to every value of x. As soon as the slope becomes a step at x_c (right panel), we do not know anymore which rank r, between $r_{\min}$ and $r_{\max}$, should correspond to x_c.

To generalize the algorithm, we can imagine the discontinuity in $F(x)$ as the limit of a very steep slope (as illustrated in figure 2). As long as there is a slope (left panel), we know which value of the rank $r = F(x)$ corresponds to every value of x: a small uncertainty in x just produces a larger uncertainty in r when the slope is steeper. As soon as the slope becomes a step (right panel), we do not know anymore which rank r, between $r_{\min}$ and $r_{\max}$, should correspond to x_c.

The solution is then to make the transformation stochastic and transform x to a random rank (with uniform distribution) between $r_{\min}$ and $r_{\max}$. In this way, the forward transformation will transform the marginal distribution of all variables to the target distribution [with cdf $G(z)$] as required, the discrete events being transformed into a continuous variable by the stochastic transformation; and the backward transformation will transform it back to a discrete event, by transforming all ranks between $r_{\min}$ and $r_{\max}$ to x_c.

1.4 Extreme events

With an ensemble of size m, there is a probability $\frac{1}{m+1}$ that the value of any variable x is above tha maximum of the ensemble, and the same probability that it is below the minimum. These extreme events are missed by the ensemble, so that the tails of the marginal distributions cannot be transformed into Gaussian tails using the method described above (which is based on the available ensemble only). If dealing adequately with extreme events (outside the range of the ensemble) is important, an additional source of information is needed.

One possible solution to this problem is to have a wider catalogue of possible realization of x (like climatological data, historical records or data from nearby space/time location, by assuming space or time homogeneity of the statistics), and use this catalogue to specify the shape of the tails of the probability distributions. For instance, the quantiles of the ensemble can be merged with the quantiles of the catalogue to obtain an extended description of the transformation function. This simple solution is implemented in the module **anatail**.

1.5 Transformation of observations

In many practical situations, if the observation operator $\mathcal{H}$ is simple enough, it may be equivalent[2] or sufficient to describe the dependence between $\mathbf{x}$ and $\mathbf{y} = \mathcal{H}(\mathbf{x})$ using the statistics of the prior ensemble. Following this assumption, we can then augment the state vector with $\mathbf{y}$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} = \mathcal{H}(\mathbf{x}) \end{bmatrix}$$

and consider $\mathbf{y}^o$ as direct observation of the component $\mathbf{y}$ of $\mathbf{X}$. Anamorphosis transformation functions are then computed for all variables of $\mathbf{X}$ including the equivalent of each observation y_j^o in $\mathbf{y}^o$.

If then, as a second assumption, the observation errors associated to every individual observations y_j^o are assumed independent, it becomes possible to transform the univariate distributions $p(y_j^o|y_j)$ to a counterpart written in terms of the transformed variable $A(y_j)$. The benefit of this transformation of the observation error probability distribution is to avoid including the backward transformation A_j^{-1} in the observation operator, which would for instance be impossible in the context of a linear observational update algorithm (since the A_j^{-1} are nonlinear by construction).

To transform $p(y_j^o|y_j)$, one possible solution is to draw a sample of the possible ranks r_{js}, $s = 1, \ldots, N$ of observation y_j^o in the prior ensemble (adequately perturbed by observation error), so that $G^{-1}(r_{js}$ provides a sample of the required transformed distribution. For each draw of the sample $s = 1, \ldots, N$, we need first to select one possible rank r_s^o for the observation error. Second, we perturb the observation equivalent of every member of the ensemble with this observation error $\tilde{y}_j^i = F_i^{-1}(r_s^o)$, where F_i is the cdf of $p(y|Hx_i)$ and i is the index of the ensemble member. And third, we compute the rank r_{js} of the observation y_j^o in the perturbed ensemble $\tilde{y}_j^i$. This algorithm can be rewritten using the anamorphic transformation defined in section 1.1, with the following steps:

- sample a rank for observation error r_s^o;
- perturb the ensemble accordingly: $\tilde{y}_j^i = F_i^{-1}(r_s^o)$;
- compute anamorphic transformation A_s^o from this perturbed ensemble;
- apply A_s^o to the observation: $A_s^o(y_j^o)$.

By iterating this algorithm on s, we obtain a sample of the transformed distribution.

If the probability distribution for observation errors $p(\epsilon = y - Hx)$ is symmetric and independent of the state x of the system, it is equivalent (and much less expensive) to perturb the observation y^o rather than the prior ensemble. The algorithm simplifies to:

- perturb observation with observation error: $y_s^o = y^o + \epsilon_s$;
- apply anamorphosis transformation to y_s^o,

where the anamorphosis transformation is computed using the observation equivalent of the prior ensemble (without perturbation).

2 Description of the modules

In this section, the modules are described one by one, giving for each of them: the method that has been implemented, the list of public variables and public routines (with a description of input and output data), the MPI parallelization, and an estimation of the computational cost as a function of the size of the problem.

[2]It is equaivalent if $\mathcal{H}$ is linear or if the observational update is performed using a linear scheme.

2.1 Module: anaqua

The purpose of this module is to compute the quantiles of the ensemble defining the anamorphosis transformation functions.

Method

The approach is to loop over all variables, sort the ensemble for each variable (with the heap sort algorithm), and compute the quantiles by linear interpolation in the sorted ensemble. Options are provided to deal with ensemble members with unequal weights (either a global weight or a different weight for every variable). In this case, the interpolation to compute the quantiles is based on the cumulated weights.

Public variables

None.

Public routines

ens_quantiles: to compute the quantiles of the ensemble:

- `qua` **(output)** : quantiles of the ensemble;
- `ens` **(input)** : ensemble;
- `quadef` **(input)** : definition of the quantiles (list of the required ranks);
- `enswei` **(input, optional)** : global weight for each ensemble member (default=equal weights);
- `ensweiloc` **(input, optional)** : local weights for each ensemble member and each variable (default=equal weights).

MPI parallelization

For ensemble with many state variables, MPI parallelization is easily obtained by making each processor work on a different part of the state vector. Since the operations on different state variables are independent, no MPI operations needed to be implemented inside the routine.

Computational cost

The computational complexity of the algorithm can be written:

$$C \sim k_1 nm \log m + k_2 n \tag{5}$$

where n is the number of state variables, m, the size of the ensemble, and k_1, k_2, are order 1 constants. The first term corresponds to the sorting of the input ensemble, and the second term to the interpolation in the sorted ensemble.

2.2 Module: anatra

The purpose of this module is to apply forward and backward anamorphosis transformation functions.

Method

The approach is to loop over all variables, localize the variable to transform among the ensemble quantiles, and linearly interpolate among the corresponding quantiles of the target distribution. The transformation deals with discrete events (non-distinct quantiles) using random transformation in the range of the corresponding quantiles.

Public variables

None.

Public routines

ana_forward: forward anamorphosis transformation of input ensemble/vector/variable:

> `ens/vct/var` **(input/output)** : ensemble/vector/variable to transform;
>
> `qua` **(input)** : ensemble quantiles (provided by module **anaqua**);
>
> `quaref` **(input)** : quantiles of the target distribution;
>
> `rank` **(input, optional)** : rank to use for discrete event (default=random).

ana_backward: backward anamorphosis transformation of input ensemble/vector/variable:

> `ens/vct/var` **(input/output)** : ensemble/vector/variable to transform;
>
> `qua` **(input)** : ensemble quantiles (provided by module **anaqua**);
>
> `quaref` **(input)** : quantiles of the target distribution.

MPI parallelization

For ensemble with many state variables, MPI parallelization is easily obtained by making each processor work on a different part of the state vector. Since the operations on different state variables are independent, no MPI operations needed to be implemented inside the routine.

Computational cost

The computational complexity of the algorithm can be written:

$$C \sim k_1 n \log q + k_2 n \tag{6}$$

where n is the number of variables to transform, q, the number of quantiles used to define the transformation, and k_1, k_2, are order 1 constants. The first term corresponds to the localization of the variable in the list of quantiles, and the second term to the interpolation between the quantiles.

2.3 Module: anaobs

The purpose of this module is to apply forward anamorphosis transformation to the observation error probability distribution.

Method

In the general case, the approach is to loop over all variables, perturb the ensemble according to observation error probability distribution (with the same rank for all members), compute the quantiles of this perturbed ensemble, and use these quantiles to transform the input observation. This operation is then repeated several times to produce a sample of the transformed observation error probability distribution.

In case of a symmetric observation error probability distribution, the approach is to loop over all variables, apply perturbation to the input observation, and apply forward anamorphosis using the quantiles of the ensemble. This operation is then repeated several times to produce a sample of the transformed observation error probability distribution.

Public variables

None. The type of observation error probability distribution ('gaussian', 'lognormal', 'gamma' or 'beta', default='gaussian') is defined through the module `obserror`.

Public routines

ana_obs: transformation of observation error probability distribution (general case):

- `anaobs` **(output)** : sample of transformed observation error probability distribution;
- `obsens` **(input)** : ensemble equivalent to observations;
- `obs` **(input)** : observation vector to be transformed;
- `obserror` **(input)** : spread of observation error (precise meaning depending on the type of distribution);
- `quadef` **(input)** : definition of the quantiles used for anamorphosis;
- `quaref` **(input)** : quantiles of the target distribution.

ana_obs_sym: transformation of observation error probability distribution (symmetric case):

- `anaobs` **(output)** : sample of transformed observation error probability distribution;
- `obs` **(input)** : observation vector to be transformed;
- `obserror` **(input)** : spread of observation error (precise meaning depending on the type of distribution);
- `obsqua` **(input)** : quantiles of ensemble equivalents to observations (provided by module **anaqua**);
- `quaref` **(input)** : quantiles of the target distribution.

MPI parallelization

For problems with many observations, MPI parallelization is easily obtained by making each processor work on a different part of the observation vector. Since the operations on different observations are independent, no MPI operations needed to be implemented inside the routine.

Computational cost

In the general case, the computational complexity of the algorithm can be written:

$$C \sim k_1 nsm + k_2 nsm \log m + k_3 ns(k_4 + \log q) \tag{7}$$

where n is the number of variables to transform, m, the size of the ensemble, s, the size of the sample to produce, q, the number of quantiles used to define the transformation, k_1, the cost

of sampling the univariate observation error probability distribution, and k_2, $k_3\,k_4$, are order 1 constants. The first term corresponds to the perturbation of the ensemble, the second term to the computation of the quantiles of the perturbed ensembles, and the third term to the transformation of the observation vector.

In case of a symmetric observation error probability distribution, this cost is reduced to:

$$C \sim k_1 ns + k_3 ns(k_4 + \log q) \tag{8}$$

The first term corresponds to the perturbations of the observation vector, and the second term to the transformation of the perturbed observation vectors.

2.4 Module: anatail

Module not yet available.

References

Brankart, J.-M., C.-E. Testut, D. Béal, M. Doron, C. Fontana, M. Meinvielle, P. Brasseur, and J. Verron, 2012: Towards an improved description of ocean uncertainties: effect of local anamorphic transformations on spatial correlations. *Ocean Science*, **8**, 121–142.