Supplement of Geosci. Model Dev., 16, 3459–3478, 2023
https://doi.org/10.5194/gmd-16-3459-2023-supplement

Geoscientific Model Development
Open Access
EGU

Supplement of

Reconstructing tephra fall deposits via ensemble-based data assimilation techniques

Leonardo Mingari et al.

Correspondence to: Leonardo Mingari (leonardo.mingari@bsc.es)

Estimation of measurement errors

The assimilation methods require a dataset of measurements along with the corresponding absolute (GNC method) or relative (GIG method) errors. Consequently, the assumptions made to establish the observation errors are critical for this work. The observation dataset include measurements spanning several orders of magnitude, typically from a fraction of millimetre to a few metre of deposit thickness. Consequently, observation error standard deviations are assumed to be dependent on the measured value.

The strategy adopted in this study to estimate measurement errors provides reasonable estimates based on a clustering algorithm. Specifically, a spectral clustering algorithm (Pedregosa et al. 2011) is used to organize the observational data into groups with similar characteristics and an absolute and relative error is assigned to each group or cluster. The error for the j-th measurement is approximated by the standard deviation associated with the corresponding cluster data. In order to estimate the relative error

$$\epsilon_j^r = \frac{\epsilon_j}{y_j^t}$$

the true value y_j^t is approximated by the cluster mean value.

The classification of observations into groups requires some way of computing the distance or the similarity between each pair of observations. All this information is gathered in the so-called affinity matrix. In this work, the affinity matrix with elements a_{ij} is constructed using the following definition:

$$a_{ij} = \exp\left(-\frac{1}{2}\, d_{ij}^2\right)$$

where d_{ij} is a non-Euclidean distance measure between the i-th and j-th observations computed according a reasonable metric. The affinity should be one for identical points, whereas for very dissimilar pairs of points the affinity should be close to zero. The following definition was adopted to compute the dimensionless distance:

$$d_{ij}^2 = \left(H_{ij}(\mathrm{km})/75\,\mathrm{km}\right)^2 + \left|\log(y_i^o/y_j^o)\right|^2 \tag{S1}$$

where H_{ij} is the geographic distance in kilometres between the i-th and the j-th observations computed using the Haversine formula and y_j^o refers to the j-th measurement of deposit thickness in centimetres. According to this definition, two observations are similar or close to each other when they have the same order of magnitude and are less than 75 km apart (a few grid cells of the computational domain). The affinity matrix after applying the clustering algorithm with 9

clusters reflects the similarity among data points in the same cluster as shown in fig. S1.

The deposit thickness measurement data grouped by clusters is also represented by a clustered box plot diagram in fig. S2. Finally, a map of the measurement sites grouped according to the clustering algorithm results is shown in fig. S3. Notice that this procedure allows us to distinguish between very proximal data in regions strongly affected by ashfall and deposit thickness measurements above 10 cm (cluster 4) and proximal data in regions moderately or not affected by ashfall (cluster 8), e.g. samples collected upwind from the volcano. Specifically, the proximal cluster 8 includes a few zero-valued observations (not shown in fig. S2).

Observational dataset splitting

The full observational dataset was split into two subsets: dataset A (for assimilation) and dataset B (for validation). The splitting procedure aims to reduce the correlation between both subsets. Nevertheless, a significant correlation is still expected as the sampling sites are distributed over similar paths.

We use an iterative procedure: starting from initial datasets A and B, we seek the most uncorrelated data on dataset B based on some dissimilarity measure; this data is removed from dataset B and inserted into the dataset A. This procedure is repeated until the desired dataset size is reached.

Next, we have to define a way to determine the "most uncorrelated" data based on a convenient dissimilarity measure. To this purpose, we define the following 1-to-N similarity measure for the j-th measurement:

$$A_j = \max_i a_{ij}$$

where a_{ij} are the elements of the affinity matrix computed on the dataset B using the non-Euclidean distance given by eq. S1. The measurement with the minimum A_j (maximum dissimilarity) is considered the most uncorrelated observation on dataset B and is reassigned to the dataset A.

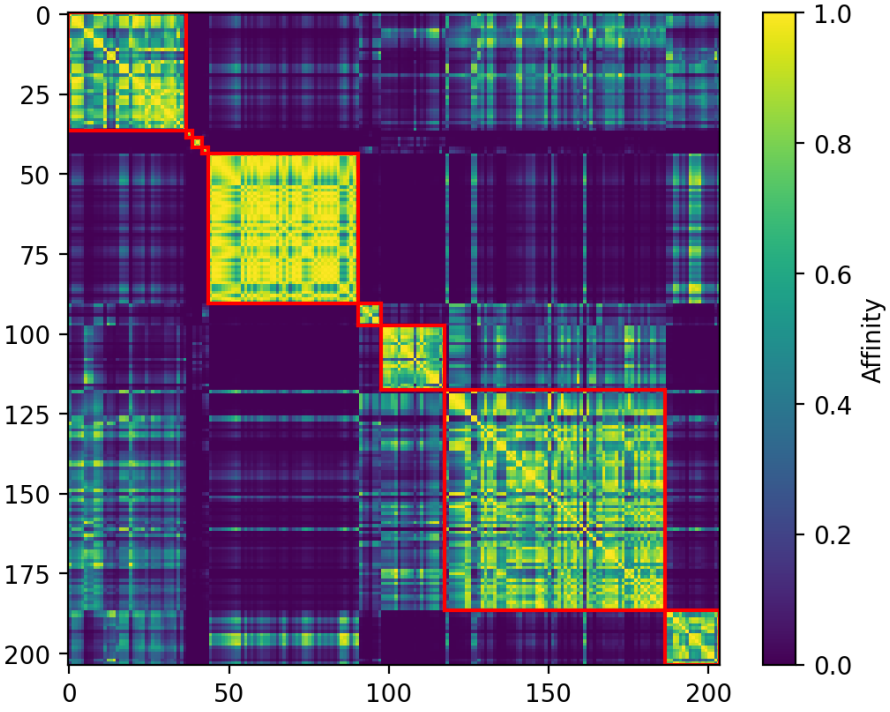

Figure S1: Affinity matrix after applying the clustering algorithm with 9 clusters.

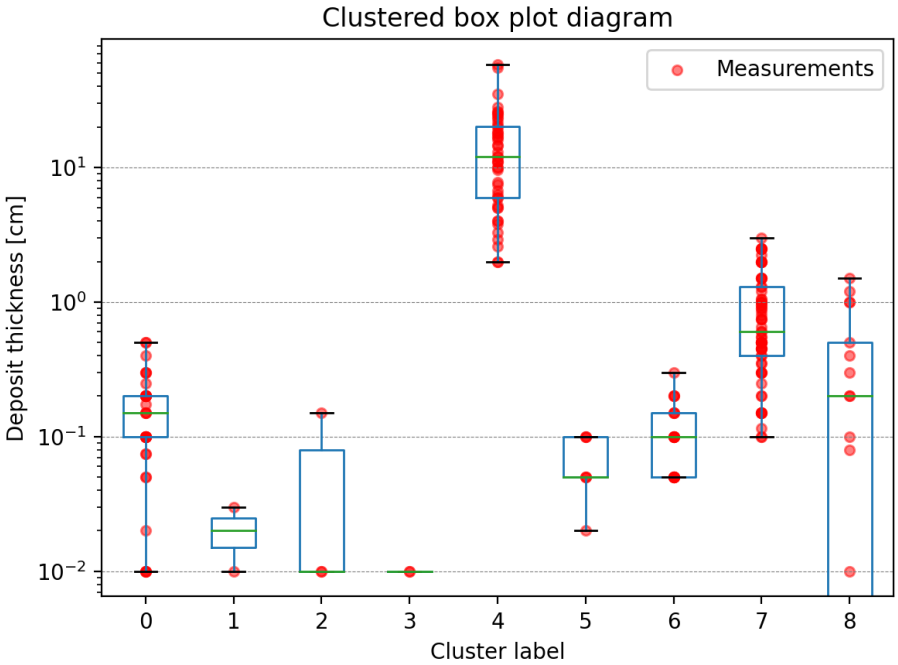

Figure S2: Clustered box plot diagram grouping the 204 measurements.

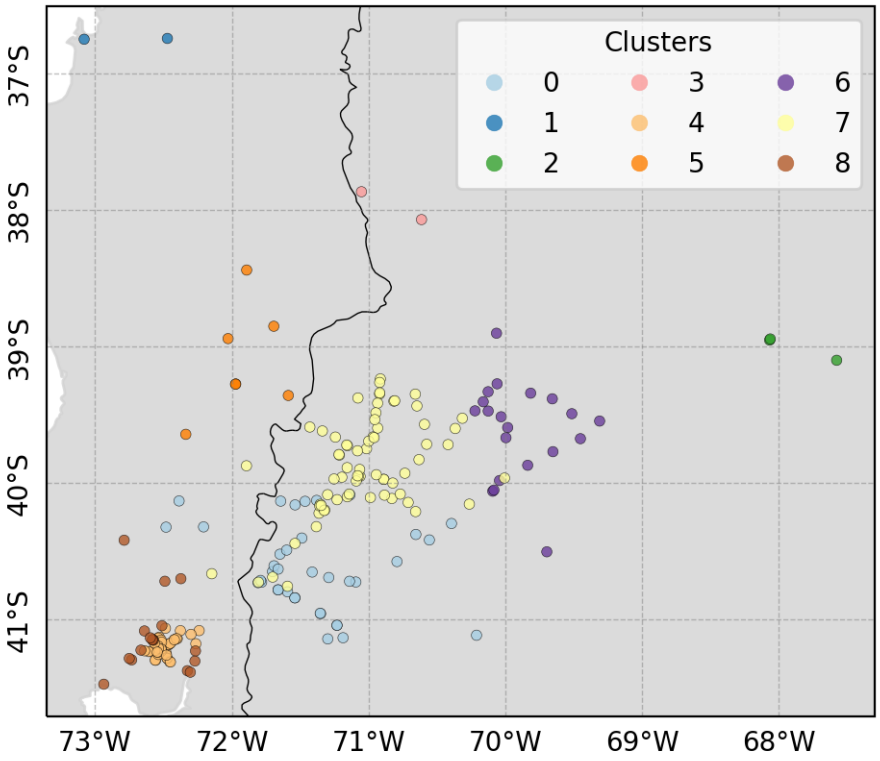

Figure S3: Map of the measurement sites grouped according to the clustering algorithm results.

Codes and datasets

Codes and datasets have been archived on Zenodo at:

DOI 10.5281/zenodo.7259531

This is the contents of the directory:

```
├── config.ini                #General configuration file
├── assimilation.py           #Module with the assimilation methods
├── method_enkf.py            #Assimilation using the EnKF method
├── method_gig.py             #Assimilation using the GIG method
├── method_gnc.py             #Assimilation using the GNC method
├── compute_metrics.py        #Compute validation metrics
├── DATA
│   ├── grl54177.csv          #Observation dataset (Van Eaton et al., 2016)
│   ├── reckziegel.csv        #Observation dataset (Recziegel, 2020)
│   ├── errors.py             #Compute error estimates
│   ├── clustering.py         #Clustering algorithm
│   ├── metrics.py            #Module with metric definitions
│   └── romero                #Isopach map (Romero et al., 2016)
│       ├── isopachs.cpg
│       ├── isopachs.dbf
│       ├── isopachs.prj
│       ├── isopachs.shp
│       └── isopachs.shx
├── plot_histograms.py        #Script to generate Fig. 4
├── plot_map.py               #Script to generate Fig. 5
├── plot_comparison.py        #Script to generate Fig. 6
├── plot_bars.py              #Script to generate Fig. 7
├── plot_mapx1.py             #Script to generate Fig. 8
├── plot_metrics.py           #Script to generate Fig. 9
└── plot_source.py            #Script to generate Fig. 10
```

References

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. "Scikit-Learn: Machine Learning in Python." *J. Mach. Learn. Res.* 12: 2825–30.