KNN-WG Formula

The KNN-WG operates by assessing the similarity of weather conditions to those of previous days. To simulate weather variables for a new day (t+1), we begin by selecting days with similar characteristics to those simulated for day t from the historical record. From this selection, one of the nearest neighbors is chosen based on a predefined probability distribution or kernel. Subsequently, the observed values for the day immediately following that nearest neighbor's day are adopted as the simulated values for day t+1 (Sharif et al., 2007). The software follows these steps, and for more in-depth information, you can refer to Sharif and Burn (2006).

Step 1: Consider the target variables for each station in the daily resolution of the historical records. If desired, you can calculate the regional means of the stations before inputting the data into the tool.

Step 2: Select the current day for analysis. If you intend to predict data for tomorrow, you should have access to the historical data for today, which will be represented as the X_t matrix.

Step 3: Choose the value for the neighbor's matrix size, denoted as L. So, you should select a temporal window of width w, which is typically set to 14 days. This temporal window encompasses one week before and one week after the current day. Therefore, if you have N years of historical data, the L size can be determined using the formula below:

$$L = N \times (W - 1) - 1$$

Step 4: Calculate the neighbor matrix, denoted as C_t. This matrix will have dimensions of L×P, where P represents the number of weather variables.

Step 5: Determine the covariance matrix, C_t, for day t by utilizing the data block with dimensions of L×p, where p signifies the number of weather variables.

Step 6: Calculate the Mahalanobis distances (as described by Davis, 1986) between the X_t vector and each vector in the neighbor matrix X_i, where i ranges from 1 to L. Both X_t and X_i have dimensions of 1×P. The Mahalanobis distance can be defined using the following formula:

$$d_i = \sqrt{(X_t - X_i) covC_t^{-1} (X_t - X_i)^T}$$

where T denotes the transpose operation, and $covC_t^{-1}$ is the inverse of the covariance matrix.

Step 7: You should choose a number (K) to select K values of Mahalanobis distances and then select one of the first K nearest neighbors. Determining the number of the first K nearest neighbors to retain for resampling out of the total L neighbors can be done using various methods. Lall and Sharma (1996) suggested using the generalized cross-validation score (GCV) to choose K. Alternatively, Rajagoplan and Lall (1999) and Yates et al. (2003) recommended a heuristic method for selecting K with this formula:

$$K = \sqrt{L}$$

Step 8: Sort the Mahalanobis distances (d_i) in ascending order and select the K nearest neighbors from the top of the sorted list. Select the ds for this array.

Step9: Calculate the weights W_j for the j^{th} neighbor and compute the cumulative probabilities P_j using the following formulas:

$$W_j = \frac{1/j}{\sum_{i=1}^{K} 1/i}$$

$$P_j = \sum_{i=1}^{j} W_i$$

Step 10: Choose a random number r between 0 and 1. If r is less than P_1, select ds_1. If r equals P_k, select ds_k. For values of j where P_{j-1} is less than r and r is less than P_j, choose ds_j.

Step 11: Locate the selected value of ds in the Mahalanobis distances array (d_i) and save the weather variables' values for the selected day as the X_{t+1} data.

Step 12: Replace X_t with X_{t+1} and repeat steps 1 to 12.

Reference: Improved K -Nearest Neighbor Weather Generating Model

What is the basis of ensemble?

In KNN-WG, we've introduced an ensemble method of nr runs. In this approach, we calculate the weight for each run and then multiply this weight by each run.

$$V^{ens} = \frac{\sum_{i=1}^{nr} \frac{V_i^{knn}}{dV_i}}{\sum_{i=1}^{nr} 1/dV_i}$$

$$dV_i = \left(\bar{V}_i^{obs} - \bar{V}_i^{calKnn}\right)^2$$

nr: Number of runs

V^{ens}: Ensemble-averaged value of Variable

V_i^{knn}: Output of KNN model in the i^th run

$\bar{V}_i^{obs}$: Mean of observed variable in the calibration period

$\bar{V}_i^{calKnn}$: Mean of Output of KNN model in the calibration period