Documentation and mathematical workings of the Data-& Shift- generator

Benedikt Stratmann

October 22, 2022

1 Goal of the Generator

Under given or not given constraints on input value ranges, input behaviour and output behaviour, synthetic data sets are to be generated. In their current state, these data sets represent regression problems of varying complexity. These data sets should also be able to integrate data drifts, i.e., time-dependent changes in the input data, as well as concept shifts, i.e., time-dependent changes in the mapping function.
In the following, the generated input features are denoted by $I \in \mathbb{R}^{N \times F}$. N denotes the number of input data points over the considered time interval T. F denotes the dimensionality of the input features. Accordingly, the concept is a mapping of the form $K : \mathbb{R}^{N \times F} \rightarrow \mathbb{R}^{N \times O}$. O denotes the dimensionality of the output, also called labels.

2 Concept function

In order to generate a concept shift, i.e. a change of the concept function K, the concept function must be parametrized.

$$K_{\Theta}(i \in I) = \sum_{k=0}^{M} \Theta_k * \psi(i)$$

$\psi(\cdot)$ is a possible, randomly chosen and initialized regression model. Θ_k denotes the influence of the k-th regression model on the concept K.

By linearly combining different models, a variety of different concepts can be generated. Currently M different regression models are chosen. The following regression algorithms are considered:

- MLPRegressor
- KernelRidgeRegressor, with a kernel, chosen from the following list:
  - polynomial
  - rbf
  - laplacian
  - sigmoid,
  - cosine
- TreeRegressor

These regression models are then each trained on 20 uniformly random data points so that each regression model ends up describing a different mapping from input features to labels.

3 Concept shift

To change the concept function at a previously defined point, we change the influence weights Θ_k. We assume at this point that both time and type of the shift as well as the target of the shift are known. If a label is requested from the concept for a time and an input feature, there are two possibilities. Possibility one: we are not in a drift, the function K_Θ is evaluated and returned. Possibility two: we are in a shift, which means depending on the **characteristic (time, type, duration** D**,** Θ_{Target}**)** of the shift the function evaluation is different:

- sudden-Shift:
  As soon as we reach the time location of the shift, a new parametrization of the linear combination applies, the evaluation is done with $K_{\Theta_{Target}}(\cdot)$
- gradual-Shift:
  For the **duration** of the shift, a transition from $K_\Theta(\cdot)$ to $K_{\Theta_{Target}}(\cdot)$ is performed. In this case, one of the two functions is randomly selected for evaluation. The probability of the selection behaves linear. At the beginning of the shift, $K_\Theta(\cdot)$ is selected with 100%, in the centre the selection is made with a chance of 50%/50% and at the end of the **duration** $K_{\Theta_{Target}}(\cdot)$ is used with a probability 100%.

- incremental-Shift:
  A transition from $K_\Theta(\cdot)$ to $K_{\Theta_{Target}}(\cdot)$ takes place over the **duration** of the shift. Unlike gradual shift, however, interpolation is performed between Θ and Θ_{target}. $\Theta_t = \frac{D-t}{D}\Theta + \frac{t}{D}\Theta_{target}$ for $0 \leq t \leq D$. So the evaluation is always done with $K_{\Theta_t}(\cdot)$

- reoccurring concept-Shifts:
  This type of shift is an extension of the previously mentioned types. Instead of just shifting to a new mapping $K_{\Theta_{Target}}(\cdot)$, this type of shift deals with first shifting to $K_{\Theta_{Target}}(\cdot)$ and then shifting back to $K_\Theta(\cdot)$. For sudden shifts this means that at the beginning of the duration until the end of the duration $K_{\Theta_{Target}}(\cdot)$ is used and otherwise $K_\Theta(\cdot)$.
  For gradual shifts, at the centre of the drift duration $K_{\Theta_{Target}}(\cdot)$ is used with a 100% probability for the evaluation and at the beginning and end of the duration $K_\Theta(\cdot)$ is used with a 100% probability. For incremental shifts we split the shift into two incremental shifts. In the first half of the duration from Θ to Θ_{Target} and in the second half from Θ_{Target} to Θ.

4 Data generation

Given the Concept function and the Concept shift, the generation of the input data I will be described in the following.
The generator is able to generate uniform, constant, gaussian and periodic features, via a sampling process, where a function maps a feature specific input vector w and a timestamp t to a value or probability distribution and returns the value or a sample drawn from that distribution.
$\boldsymbol{w}$ can be viewed as the parametrization of the sampling process. As an example the code of the uniform sampling method is given in algorithm 1.

Algorithm 1 dist_uniform(w,t)

$interval \leftarrow random(0,1) \cdot w[1]$	▷ w[1] is used as the interval
$return \leftarrow interval - w[1]/2 + w[0]$	▷ w[0] is used as the centre

Other distributions are constructed similarly and may use more or less entries of w and use the additional time information provided by t. As we want to later introduce Datadrifts, the sampling process has to be parametrized. As different distributions or functions have different characteristic markers this parametrization is dealt with by using w.

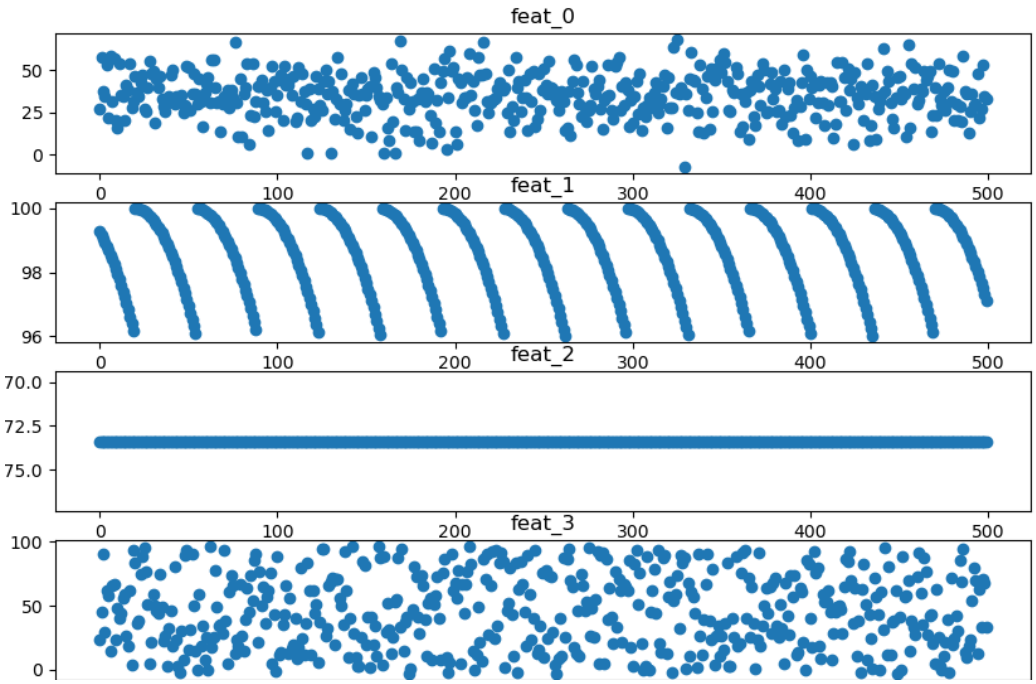

Figure 1: Examples to each of the different distributions, where the x-axis denotes the time and the y-axis denotes the values of the features.
feat_0 sampled from a gaussian distribution
feat_1 sampled from a periodic distribution
feat_2 sampled from a constant distribution
feat_3 sampled from a uniform distribution

Similar function characteristics are assigned to the same index of w.

After the basic features are generated, dependent and correlated features are generated or built out of the already present features.
Dependent features are built by again training two models for regression on random data, therefore building more or less random mappings. Then two already existing features are chosen as input for the models and the output is summed with weights determined by w.

Correlation relies on generating a random positive definite correlation matrix and recomputing the features that are to be correlated with each other. A possible solution for generating such a matrix can be seen in Davies and Higham [2000].

5 Data drift

The realization of Data drifts is handled similarly to section 3. Instead of changing the weights of the linear combination , the w vector that defines the behaviour of the distributions is changed, depending on the class of drift. Therefore, if a feature is marked to drift at a certain point with a certain duration, the w vector of that feature will be changed just as Θ was changed for the concept drifts.

6 Noise

Noise can be added to the features and outputs of the generator. The Noise is however added after the generation. The noise can therefore be interpreted as sensor readout variance.
Given the generated value x and a random variable $z \sim \mathcal{N}(0, \sigma_{noise})$ the new noisy value y is:

$$y = x \cdot (z + 1)$$

References

Davies, P. I. and Higham, N. J. (2000). Numerically stable generation of correlation matrices and their factors. *BIT Numerical Mathematics*, 40:640–651.