Choosing a Discretization Algorithm

Question

How do should I choose an appropriate discretization algorithm if I don't have any knowledge about the domain?

Answer

It depends on the type of model you are developing:

- For models with a target variable, follow the discretization approach for **Supervised Learning**.
- For models without a target variable, i.e. with all variables having (a priori) equal importance, follow the discretization approach for **Unsupervised Learning**.

Discretization for Supervised Learning

In the context of **Supervised Learning**, the **Tree** algorithm is generally the best approach. All continuous variables will be binned based on their individual relationship with the target variable.

⚠️ Please note that the **Tree** discretization only works with *discrete* target variables. However, if your target variable is *continuous*, you will first need to discretize it individually using the **Manual** approach. This way you can discretize the target based on your own knowledge. Alternatively, you can utilize the automatic unsupervised discretization algorithms (e.g. density approximation). Once the thresholds are defined, the target variable is considered *discrete*, and then it can be selected as a target for the **Tree** algorithm.

If the **Tree** discretization fails (which means that there is no significant relationship between the variable and the target), the discretization will default to the algorithm for **Unsupervised Learning** (see below).

Discretization for Unsupervised Learning

Unless you override this default sequence, BayesiaLab utilizes its discretization algorithms in the following order:
1. Density Approximation, *then, if this algorithm fails*

2. K-Means, *then, if this algorithm fails*

3. Normalized Equal Distances, *then, if this algorithm fails*

4. Equal Distances

We do not recommend **Equal Frequencies** for the following two reasons:

- The initial density function is lost.
- The marginal entropy of the discretized variable is at its maximum value. With the objective of BayesiaLab's structural learning algorithms being the reduction of the global entropy of the system, the **Equal Frequencies** discretization tool leads to overly complicated networks.

**Number of Bins**

BayesiaLab's structural learning algorithms are based on the Minimum Description Length score (MDL). The general idea is to add a link between two variables *only* if the relationship is strong enough to compensate for the increased structural complexity due to the conditional probability that quantifies the nodes' relationship. The more intervals there are per variable, the more complex are the resulting conditional probability tables. Hence, more data is needed to find relationships strong enough to compensate the complexity.

Thus, the maximum number of intervals to choose depends on the size of the dataset you are analyzing. Defining too many intervals can lead to too sparse networks. Defining too few intervals can lead to overly complex networks.

- For datasets with 1000 or more observations, choosing five bins is usually a safe choice.
- At least three bins are necessary to capture nonlinear relationships.
- Changing the Structural Coefficient ( ) allows changing the balance in the MDL score between relation strength and structural complexity.