# installation   
#install.packages("daltoolbox")  
  
# loading DAL  
library(daltoolbox)

About the technique - smoothing\_cluster: discretization/smoothing by defining bins via clustering instead of fixed intervals.

# Discretization and smoothing

Discretization transforms continuous functions, models, variables, and equations into discrete counterparts.

Smoothing creates an approximating function to capture important patterns while reducing noise or high-frequency variation.

Defining bin intervals is an important step to enable the approximation.

# General function to evaluate different smoothing techniques

Sample data (iris) to illustrate clustering-based discretization/smoothing.

iris <- datasets::iris  
head(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 1.4 0.2 setosa  
## 3 4.7 3.2 1.3 0.2 setosa  
## 4 4.6 3.1 1.5 0.2 setosa  
## 5 5.0 3.6 1.4 0.2 setosa  
## 6 5.4 3.9 1.7 0.4 setosa

Apply clustering-based smoothing and inspect bins.

# smoothing using clustering  
obj <- smoothing\_cluster(n = 2)   
obj <- fit(obj, iris$Sepal.Length)  
sl.bi <- transform(obj, iris$Sepal.Length)  
print(table(sl.bi))

## sl.bi  
## 5.19875 6.58   
## 80 70

obj$interval

## [1] 4.300000 5.889375 7.900000

Evaluate conditional entropy between bins and species.

entro <- evaluate(obj, as.factor(names(sl.bi)), iris$Species)  
print(entro$entropy)

## [1] 1.097573

# Optimizing the number of binnings

Optimize the number of bins (search 1:20) and refit.

opt\_obj <- smoothing\_cluster(n=1:20)  
obj <- fit(opt\_obj, iris$Sepal.Length)  
obj$n

## [1] 8

obj <- fit(obj, iris$Sepal.Length)  
sl.bi <- transform(obj, iris$Sepal.Length)  
print(table(sl.bi))

## sl.bi  
## 4.4 4.70909090909091 5.04666666666667 5.63243243243243   
## 5 11 30 37   
## 6.215625 6.65 7.06 7.67142857142857   
## 32 18 10 7