Monotonic Optimal Binning in Credit Risk

This project mainly implements the Monotonic Optimal Binning(MOB) algorithm in SAS 9.4. We extend the application of this algorithm which can be applied to numerical and categorical data. In order to avoid the problem of creating too many bins, we optimize the p-value iteratively and provide `bins size first binning`, `monotonicity first binning`, and `chi merge binning` methods for users to discretize data more conveniently.

How to use

Step 1. Download this repository

```
git clone https://github.com/cdfq384903/MonotonicOptimalBinning.git
```

Step 2. Upload source code and required data

1. Upload source code as the frame shown below.

 folder structure 2

> Note: we had made some modifications to the dataset `german_data_credit_cat.csv`. Details are shown below :

1. Rename all columns
2. Change the value of column `Cost Matrix(Risk)` :

Types of Credit Risk	original value	Revised value
Good Risk	1	0
Bad Risk	2	1

Step 3. Usage Demo

Numerical variables

Initialize parameters:

```
%let data_table = german_credit_card;
%let y = CostMatrixRisk;
%let x = AgeInYears CreditAmount DurationInMonth;
%let exclude_condi = < -99999999;
%let min_samples = %sysevalf(1000 * 0.05);
%let min_bads = 10;
%let min_pvalue = 0.35;
%let show_woe_plot = 1;
```

```
%let lib_name = TMPWOE;
%let is_using_encoding_var = 1;
```

Size First Binning(SFB)

Run `MainSizeFirstBining.sas` script

```
 %let min_bins = 3;
 %let max_samples = %sysevalf(1000 * 0.4);

 %init(data_table = &data_table., y = &y., x = &x., exclude_condi =
&exclude_condi.,
       min_samples = &min_samples., min_bads = &min_bads., min_pvalue =
&min_pvalue.,
       show_woe_plot = &show_woe_plot.,
       is_using_encoding_var = &is_using_encoding_var., lib_name =
&lib_name.);
 %initSizeFirstBining(max_samples = &max_samples., min_bins = &min_bins.,
max_bins = 7);
 %runMob();
```

SFB RESULT OUTPUT - `DurationInMonth`:

> Note: The image above shows the Woe Transformation Result of variable `DurationInMonth` with applying `SFB Algorithm`. It clearly presents the monotonicity of the WoE value.

SFB RESULT OUTPUT - `CreditAmount` :

> Note: The image above shows the Woe Transformation Result of variable `CreditAmount` with applying `SFB Algorithm`. It violates the monotonicity of WoE because `SBF Algorithm` will tend to meet the bins relevant restrictions as priority.

Monotonic First Bining(MFB)

Run `MainMonotonicFirstBining.sas` script

```
%init(data_table = &data_table., y = &y., x = &x., exclude_condi =
&exclude_condi.,
      min_samples = &min_samples., min_bads = &min_bads., min_pvalue =
&min_pvalue.,
      show_woe_plot = &show_woe_plot.,
      is_using_encoding_var = &is_using_encoding_var., lib_name =
&lib_name.);
%initMonotonicFirstBining();
%runMob();
```

MFB RESULT OUTPUT - `DurationInMonth`:

> Note: The image above shows the Woe Transformation Result of variable `DurationInMonth` with applying `MFB Algorithm`. It presents the monotonicity of WoE.

MFB RESULT OUTPUT - `CreditAmount` :

> Note: The image above shows the Woe Transformation Result of variable `CreditAmount` with applying `MFB Algorithm`. It presents the monotonicity of WoE, but it is likely to lead to some issues such as excessive sample proportion or an insufficient number of bins or bins size.

Categorical variable

Initialize parameters:

```
%let data_table = german_credit_card;
%let y = CostMatrixRisk;
%let x = Purpose;
%let max_bins_threshold = 30 ;
%let min_bins = 4 ;
%let max_bins = 6 ;
%let min_samples = 0.05 ;
%let max_samples = 0.4 ;
%let p_value_threshold = 0.35 ;
%let libName = TMPWOE ;
```

Chi Merge Binning (CMB)

Chi Merge Binning (CMB) is an auto binning algorithm applying chi-squared test for the merging criterion. It is also limited by the same restrictions as the SFB and MFB on bins amount, bins size, sample size, etc. Currently, the CMB cannot deal with the catergorical varibales with order.

Run `MainChiMerge.sas` script

```
%runChiMerge( dataFrame = german_credit_card, x = &x., y = &y.,
              max_bins_threshold = &max_bins_threshold.,
              min_bins = &min_bins., max_bins = &max_bins.,
              min_samples = &min_samples., max_samples = &max_samples.,
              p_value_threshold = &p_value_threshold.,
              libName = &libName.) ;
```

CMB OUTPUT RESULT :

The result of `CMB` is shown above. We can see that the `CMB Algorithm` merges the categorical variable `Purpose` in `german_credit_card` from 10 attributes to 6 groups eventually.

Macro Arguments Reference

The MonotonicOptimalBining core class

MOB algorithm macro - MFB/SFB

`MFB Algorithm` macro example:

```
%init(data_table, y, x, exclude_condi, min_samples, min_bads, min_pvalue,
      show_woe_plot, is_using_encoding_var, lib_name);
%initMonotonicFirstBining();
%runMob();
```

`SFB Algorithm` macro example:

```
%init(data_table, y, x, exclude_condi, min_samples, min_bads, min_pvalue,
      show_woe_plot , is_using_encoding_var , lib_name );
%initSizeFirstBining(max_samples , min_bins , max_bins);
%runMob();
```

Arguments

1. **`data_table`**
   Default: None
   Suggestion: a training data set.
   The `data_table` argument defines the input data set. The datasets must includes all independent variables and the target variable (response variable). For example, in `MainMonotonicFirstBining.sas` script you can pass `german_credit_card` as the given dataset which is a table structure created by `%readCsvFile()` macro.

2. **`y`**
   Default: None
   Suggestion: The label name of response variable.
   The `y` argument defines the column name of the response variable. For example, in `MainMonotonicFirstBining.sas` script you can pass `CostMatrixRisk` which exists in the dataset `german_credit_card`.

3. **`x`**
   Default: None

Suggestion: The column names of the variable for executing the alogorithm.
The `x` argument defines the column names of the chosen variables. Multiuple columns can be passed simultaneously. For example, in `MainMonotonicFirstBining.sas` script you can pass `AgeInYears CreditAmount DurationInMonth` which all exist in the dataset `german_credit_card`.

4. **`exclude_condi`**
   Default: None
   Suggestion: The condition given to exclude the observations in the variables.
   The `exclude_condi` argument defines the conditiont to exclude the observations that meet the specified condition of the variables. For example, in `MainMonotonicFirstBining.sas` script you can pass `< -99999999`, which means that the algorithm will exclude the observations that the value of the variable is less then -99999999.

5. **`min_samples`**
   Default: None
   Suggestion: The minimum sample amount that will be kept in each bin. Usually `min_samples` is suggested to be 5% of the total population.
   The `min_samples` argument defines the minimum sample that will be kept in each bin. For example, in `MainMonotonicFirstBining.sas` script you can pass `%sysevalf(1000 * 0.05)`, which means the minimum samples will be constrained by 5% of total samples (1000 obs).

6. **`min_bads`**
   Default: None
   Suggestion: The minimum positive event amount (default/bad in risk analysis) that will be kept in each bin. Usually `min_bads` is suggested to be 1.
   The `min_bads` argument defines the minimum positive event amount that will be kept in each bin. For example, in `MainMonotonicFirstBining.sas` script you can pass 10, which means that the minimum bads will be constrained by a minimum of 10 positive events in each bins.

7. **`min_pvalue`**
   Default: None
   Suggestion: The minimum threshold of p-value for the algorithm to decide whether merge the two bins or not. Usually a higher `min_pvalue`, the algorithm will reduce the times of merging bins.
   The `min_pvalue` argument defines the minimum threshold of p value. For example, in `MainMonotonicFirstBining.sas` script you can pass 0.35, which means that the alogorithm will decide to merge the two bins if the p-value of the statistical test (Z-Test) conducted between them is greater than 0.35. The argument will iteratively decrease its value if there is no p-value of the statistical test (Z-Test) conducted between any two bins greater than the given parameter and the final bins amount is still greater than `max_bins`.

8. **`show_woe_plot`**
   Default: None
   Suggestion: Boolean(0, 1) : Whether showing the woe plot when MOB algorithm is running.
   The `show_woe_plot` argument defines whether showing the woe plot in the algorithm process or not. For example, in `MainMonotonicFirstBining.sas` script you can pass 1, which means that the SAS will show the woe plot result for each given `x`.

9. **`is_using_encoding_var`**
   Default: None
   Suggestion: The boolean(0, 1) of using encoding var table. If your length of label name(x or y) is too long for sas macro, suggest you should open this parameter.
   The `is_using_encoding_var` argument defines the boolean(0, 1) of using encoding var table. For example, in MainMonotonicFirstBining.sas script you can try 1, which means the attributes name of data will be changed to be encoding variable.

10. **`lib_name`**
    Default: None
    Suggestion: The library name to store the output tables. If no preference, please pass `work`, which means a temporary library in SAS.
    The `lib_name` argument defines the output library name for storing tables created by the algorithm. For example, in `MainMonotonicFirstBining.sas` script you can pass `TMPWOE` which are assigned by `LIBNAME TMPWOE "/home/u60021675/output"` under the given direction.

11. **`max_samples`**
    Default: None
    Suggestion: Only use in `%initSizeFirstBining()` macro. The maximum sample will be kept in each bins. Usually `max_sample` suggest to be 40% of population to avoid a serious concentration issue on WoE binning.
    The `max_samples` argument defines the maximum sample amount that will be kept in each bin. For example, in `MainSizeFirstBining.sas` script you can pass with `%sysevalf(1000 * 0.4)`, which means the maximum samples will be constrained by a maximum limitation of observations which is 40% of population in each bins.

12. **`min_bins`**
    Default: None
    Suggestion: Only use in `%initSizeFirstBining()` macro. The minimum bins will be kept in the final woe summary output for each given `x`.
    The `min_bins` argument defines the minimum bins amount that will be kept in the final woe summary output for each given `x`. For example, in `MainSizeFirstBining.sas` script you can pass `3`, which means the algorithm will create at least 3 bins for the given `x` in each.

13. **`max_bins`**
    Default: None
    Suggestion: Only use in `%initSizeFirstBining()` macro. The maximum bins will be kept in the final woe summary output for each given `x`. Note that `max_bins` must be higher than `min_bins`.
    The `max_bins` argument defines the maximum bins amount that will be kept in the final woe summary output for each given `x`. For example, in `MainSizeFirstBining.sas` script you can pass `7`, which means the algorithm will create at most 7 bins for the given `x` in each.

Output

1. The output files created by MOB algorithm.

2. The woe summary result table created by MOB algorithm.

Print WoE result

`%printWithoutCname()` macro example:

```
%printWithoutCname(lib_name);
```

Arguments

1. **`lib_name`**
   Default: None
   Suggestion: The library which will be assigned for storing the woe summary result.
   The `lib_name` argument defines the library which will be assigned for storing woe summary result. For example, in `MainMonotonicFirstBining.sas` script you can pass `TMPWOE`, which means that the `%printWithoutCname()` macro will output the files and result table to `TMPWOE` library assigned by `LIBNAME TMPWOE(/home/u60021675/output) ;`.

Output

The output of runing `%printWithoutCname()` macro. It shows the result of all variable which was discretized.

Generate the IV summary table

`%getIvPerVar()` macro example:

```
%getIvPerVar(lib_name, min_iv, min_obs_rate, max_obs_rate, min_bin_size,
max_bin_size, min_bad_count);
```

Arguments

1. **`lib_name`**
   Default: None
   Suggestion: The library which will be assigned for storing the IV summary result.
   The `lib_name` argument defines the library which will be assigned for storing the IV summary result. For example, in `MainMonotonicFirstBining.sas` script you can pass `TMPWOE`, which means that the `%printWithoutCname()` macro will output the files and result table to `TMPWOE` library assigned by `LIBNAME TMPWOE(/home/u60021675/output) ;`.

2. **`min_iv`**
   Default: None
   Suggestion: The minimum threshold of information value (IV). Usually greater than 0.1.
   The `min_iv` argument defines the minimum threshold of the information value (IV). For example, in

`MainMonotonicFirstBining.sas` script you can pass 0.1, which means the `%getIvPerVar()` macro will mark `is_iv_pass` as 1 if IV is greater than 0.1.

3. **`min_obs_rate`**
   Default: None
   Suggestion: The minimum threshold of observation rate. `0.05` is usually given based on experiences.
   The `min_obs_rate` argument defines minimum threshold of observation rate. For example, in MainMonotonicFirstBining.sas script you can pass 0.05, which means the `%getIvPerVar()` macro will mark `is_obs_pass` as 1 if the value is greater than 0.05 and lower than `max_obs_rate`.

4. **`max_obs_rate`**
   Default: None
   Suggestion: The maximum threshold of observation rate. `0.4` is usually given based on experiences.
   The `max_obs_rate` argument defines maximum threshold of observation rate. For example, in `MainMonotonicFirstBining.sas` script you can pass 0.4, which means the `%getIvPerVar()` macro will mark `is_obs_pass` as 1 if the value is less than 0.4 and greater than `min_obs_rate`.

5. **`min_bin_size`**
   Default: None
   Suggestion: The minimum threshold of bins size. Usually set at 3.
   The `min_bin_size` argument defines the minimum amount of bins. For example, in `MainMonotonicFirstBining.sas` script you can pass 3, which means the `%getIvPerVar()` macro will mark `is_bin_pass` as 1 if the value is higher than 3 and lower than `max_bin_size`.

6. **`max_bin_size`**
   Default: None
   Suggestion: The maximum threshold of bins size. Usually set at 6.
   The `max_bin_size` argument defines the maximum amount of bins. For example, in `MainMonotonicFirstBining.sas` script you can pass 10, which means the `%getIvPerVar()` macro will mark `is_bin_pass` as 1 if the value is less than 6 and greater than `min_bin_size`.

7. **`min_bad_count`**
   Default: None
   Suggestion: The minimum number threshold of the positive events (default/bad). Usually set at 1.
   The `min_bad_count` argument defines the minimum number threshold of the positive events, defualt or bad event is commonly seen in risk analysis. For example, in `MainMonotonicFirstBining.sas` script you can pass 1, which means the `%getIvPerVar()` macro will mark `is_bad_count_pass` as 1 if the value is higher than 1.

Output

The output of `%getIvPerVar()` macro. It shows the IV information for all discretized variables.

1. `iv`: the information value per each discretized variable.
2. `is_iv_pass`: true(1) if IV higher than `min_iv` else than false(0).
3. `is_obs_pass`: true(1) if observation rate between `min_obs_rate` and `max_obs_rate` else then false(0).
4. `is_bad_count_pass`: true(1) if bad count higher than `min_bad_count` else then false(0).
5. `is_bin_pass`: true(1) if bin size between `min_bin_size` and `max_bin_size` else then false(0).

6. `is_woe_pass`: true(1) if the value of WoE have monotonicity properties else then false(0).
7. `woe_dir`: `asc` if the WoE value show a monotone increasing pattern, while `desc` if the WoE value show a monotone decreasing pattern. Otherwise, null is given.

Print WoE bar chart via IV summary filter

`%printWoeBarLineChart()` macro example:

```
%printWoeBarLineChart(lib_name, min_iv);
```

Arguments

1. **`lib_name`**
   Default: None
   Suggestion: The library which will be assigned for the data to print WoE bar chart.
   The `lib_name` argument defines the library used to store the data for plotting. For example, in `MainMonotonicFirstBining.sas` script you can pass `TMPWOE`, which means that the `%printWithoutCname()` macro will output the files and result table to `TMPWOE` library assigned by `LIBNAME TMPWOE(/home/u60021675/output) ;`.

2. **`min_iv`**
   Default: None
   Suggestion: The minimum threshold of information value. Usually set more higher than 0.1.
   The `min_iv` argument defines the minimum threshold of information value. For example, in `MainMonotonicFirstBining.sas` script you can pass 0.1, which means the `%printWoeBarLineChart()` macro will show the woe bar chart of the varibale if its IV is greater than 0.1.

Output

The output of runing `%printWoeBarLineChart()` macro. It shows the woe bar charts of the variables whose IV is greater than `min_iv`.

Generate split rule

`%exportSplitRule()` macro example:

```
%exportSplitRule(lib_name, output_file);
```

Arguments

1. **`lib_name`**
   Default: None
   Suggestion: The library which is assigned to store the split rule exported by the macro.
   The `lib_name` argument defines the library which is assigned to store the split rule exported by the macro. For example, in `MainMonotonicFirstBining.sas` script you can pass `TMPWOE`, which means that the `%printWithoutCname()` macro will output the files and result table to `TMPWOE` library assigned by `LIBNAME TMPWOE(/home/u60021675/output) ;`.

2. **`output_file`**
   Default: None
   Suggestion: The output file path which will be export split rule.
   The `output_file` argument defines the output file path which will be export split rule. For example, in `MainMonotonicFirstBining.sas` script you can try `/home/u60021675/output/`, which means the `%exportSplitRule()` macro will export the split rule to "/home/u60021675/output/" directory. Note that you DON'T need to quote the direction.

Output

The output of `%exportSplitRule()` macro.

Clear useless data table

`%cleanBinsDetail()` macro example:

```
%cleanBinsDetail(bins_lib);
```

Arguments

1. **`bins_lib`**
   Default: None
   Suggestion: The library used to store files created from the algorithm process and will be cleared eventually. Suggest to use the same value assigned in `%init()` macro.
   The `bins_lib` argument defines the library which the files in it will be cleared at the end. For example, in `MainMonotonicFirstBining.sas` script you can pass `TMPWOE`, which means bins summary files and exclude files will be deleted.

Output

The output of runing `%cleanBinsDetail()` macro. It shows the bins_summary and exclude file was be deleted.

Categorical variables binning macro - CMB

`CMB Algorithm` macro example:

```
%runChiMerge( dataFrame = german_credit_card, x = &x., y = &y.,
              max_bins_threshold = &max_bins_threshold.,
              min_bins = &min_bins., max_bins = &max_bins.,
              min_samples = &min_samples., max_samples = &max_samples.,
              p_value_threshold = &p_value_threshold.,
              libName = &libName.) ;
```

Arguments

1. **`dataFrame`**
   Default: None
   Suggestion: a training data set.
   The `dataFrame` argument defines the input data set. The datasets must includes all independent variables and the target variable (response variable). For example, in `MainChiMerge.sas` script you can pass `german_credit_card` as the given dataset which is a table structure created by `%readCsvFile()` macro.

2. **`y`**
   Default: None
   Suggestion: The label name of response variable.
   The `y` argument defines the column name of the response variable. For example, in `MainChiMerge.sas` script you can pass `CostMatrixRisk` which exists in the dataset `german_credit_card`.

3. **`x`**
   Default: None
   Suggestion: The column names of the variable for executing the alogorithm.
   The `x` argument defines the column names of the chosen variables. Multiuple columns can be passed simultaneously. For example, in `MainChiMerge.sas` script you can pass `Purpose` which exists in the dataset `german_credit_card`.

4. **`max_bins_threshold`**
   Default: None
   Suggestion: Maximum initial attributes of a variable to run CMB algorithm.
   The `max_bins_threshold` argument defines that the maximum for conducting the CMB algorithm, if the inital unique attributes of the given `x` exceed the given parameter of `max_bins_threshold` then the algorithm will stop the execution. For example, in `MainChiMerge.sas` script, you can pass `20`, which means that if the given `x` has unique attributes greater than 20, then the algorithm will stop executing.

5. **`min_bins`**
   Default: None
   Suggestion: The minimum bins will be kept in the final woe summary output for each given `x`.
   The `min_bins` argument defines the minimum bins amount that will be kept in the final woe summary output for each given `x`. For example, in `MainChiMerge.sas` script you can pass `3`, which means the algorithm will create at least 3 bins for the given `x` in each.

6. **max_bins**
   Default: None
   Suggestion: The maximum bins will be kept in the final woe summary output for each given x. Note that max_bins must be higher than min_bins.
   The max_bins argument defines the maximum bins amount that will be kept in the final woe summary output for each given x. For example, in MainChiMerge.sas script you can pass 7, which means the algorithm will create at most 7 bins for the given x in each.

7. **min_samples**
   Default: None
   Suggestion: Integer or float : The minimum sample amount that will be kept in each bin. Usually min_samples is suggested to be 5% of the total population.
   The min_samples argument defines the minimum sample that will be kept in each bin. If the given value is between 0 and 1, which means 0 < min_samples < 1, then the program will calculate the given proportion samples of the total population. For example, in MainChiMerge.sas script you can pass 0.05, which means the minimum samples will be constrained by 5% of total samples automatically calculated in the program. Or, the parameter can be passed %sysevalf(1000 * 0.05) ;, which means the minimum sample will directly be constrained as 50.

8. **max_samples**
   Default: None
   Suggestion: Integer or float : The maximum sample will be kept in each bins. Usually max_sample suggest to be 40% of the total population to avoid a serious concentration issue on WoE binning.
   The max_samples argument defines the maximum sample amount that will be kept in each bin. For example, in MainChiMerge.sas script you can pass 0.4, which means the minimum samples will be constrained by 40% of total samples automatically calculated in the program. Or, the parameter can be passed %sysevalf(1000 * 0.4), which means the maximum samples will directly be constrained as 400.

9. **p_value_threshold**
   Default: None
   Suggestion: The minimum threshold of p-value for the algorithm to decide whether merge the two bins or not. Usually a higher min_pvalue, the algorithm will reduce the times of merging bins.
   The p_value_threshold argument defines the minimum threshold of p value. For example, in MainChiMerge.sas script you can pass 0.35, which means that the alogorithm will decide to merge the two bins if the p-value of the statistical test (Chi-Squared Test) conducted between them is greater than 0.35. The argument will iteratively decrease its value if there is no p-value of the statistical test (Chi-Squared Test) conducted between any two bins greater than the given parameter and the final bins amount is still greater than max_bins.

10. **libName**
    Default: None
    Suggestion: The library which will store the woe summary result and other tables.
    The libName argument defines the library which will be loaded and show IV summary result. For example, in MainMonotonicFirstBining.sas script you can pass TMPWOE, which means that the %printWithoutCname() macro will output the files and result table to TMPWOE library assigned by LIBNAME TMPWOE(/home/u60021675/output) ;.

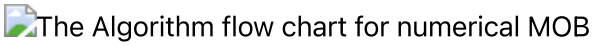
Output

1. The output files created by CMB algorithm.

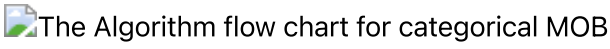
The final output of the woe binning result is stored in `woe_summary_<x>.sas7bdat`. Details are shown below:

Monotonic Optimal Bining Algorithm Flow Chart

Numerical variables



Categorical variables



Environment

SAS Studio 3.8 with SAS 9.4

References

1. German Credit Risk Analysis : Beginner's Guide . (2022). Retrieved 9 June 2022, from Kaggle
2. Pavel Mironchyk and Viktor Tchistiakov. "Monotone optimal binning algorithm for credit risk modeling.". (2017): 1-15. citation
3. SAS OnDemand for Academics. (2022). Retrieved 9 June 2022

Authors

1. Darren Tsai(https://www.linkedin.com/in/darren-yucheng-tsai/)
2. Denny Chen(https://www.linkedin.com/in/dennychen-tahung/)
3. Thea Chan(yahui0219@gmail.com)