%MSM: SAS software for survival analysis using marginal structural models

Roger Logan, Eric Tchetgen, Miguel A. Hernan

December 10, 2014

Abstract

%MSM is a set of SAS macros for survival analysis with a time-varying treatment. %MSM estimates the parameters of a marginal structural Cox model by fitting a weighted pooled logistic model using inverse probability weights for treatment and censoring. In addition %MSM can estimate absolute risks under two static interventions.

This document describes the macro options and several example runs that may be helpful in learning how the macro works for various settings of interest. A quick introduction to the MSM macro can be found in the file msm_samples.sas.

The use of this macro requires the STAT and IML packages in addition to SAS BASE.

Keywords: SAS, macro, marginal structural models

Contents

1 Macro parameters

Users need to specify the following parameters when calling the %MSM macro:

data=	Data set name.
id=	Subject unique identifier (variable name).
time=	Time of follow-up (variable name) first observation for each subject must be time=0.(In descriptions below denote time index with m.)

Structural Model Settings

outcMSM=	Time-varying outcome variable 1: event, 0: no event; missing if cens=1 or cens2=1 or cens3=1.
AMSM=	Time-varying exposure variable(s).
AMSM_class=	Categorical variables in AMSM (other than two levels). Default reference level will be first level.
covMSMbv=	Baseline variables to include in weighted models.
covMSMbh=	Baseline hazard variables (e.g., splines for time) to include in weighted models.
contMSMbh = 1	All variables listed in baseline hazard variables are continuous.
user_function_of_time = 0	This allows the user to specify their own functions of time for variables listed in covMSMbh used in the outcome model and in the calculation of the cumulative incidence, Variables should be listed in covMSMbh. When user_function_of_time = 1, the user will need to supply a macro called %user_function_of_time that creates all variables listed in covMSMbh which can be functions of the time variable and any baseline covariate listed in covMSMbv.
time_knots =	If non-missing, these are used for creating covMSMbh time categorical/ spline variables for final model and curves submacro.
classMSMbv =	Categorical variables in covMSMbv.
classMSMbh =	Categorical variables in covMSMbh.
inter_MSM=	Used to form interaction terms between all variables listed in AMSM to be used in the weighted outcome model. The msm macro will build any interactions internally so that when survival_curves = 1 the interactions are created in a consistent way.
inter_time=	Used to form interaction terms with variables listed in covMSMbh. All interactions between treatmnet and time should be included in the inter_MSM variable instead of inter_time since these interactions will also be included in the natrual course outcome model (when use_natural_course = 1). In this case all the variables listed in covMSMbh should be included in inter_MSM.

Settings for treatment and possible censoring weight models.

A=	Time-varying outcome variable 1: treated, 0: untreated if missing, weights set to 1.
covAd=	Baseline and time-varying variables in treatment model for denominator of weights.
classAd=	List of categorical variables in covAd.
covAn=	Baseline variables in treatment model for numerator of weights.(Anything listed here also needs to be listed in covMSMbv.)
classAn=	List of categorical variables in covAn.

eligible=	Eligibility variable for treatment 1: yes, 0: no If 0 then pA_d and pA_n will be set to 1.
A_censoring=	Trreatment variables to include in the censoring models. When left blank, the treatment variable listed in A will be used in all numerator and denominator models for censoring. When non-blank, variables contained in A_censoring will be added to both the numerator and denominator censoring models.
Cens=	Time-varying censoring indicator. 1: yes, 0: no
covCd=	Baseline and time-varying variables in model for denominator of weights, no treatment variables should be included in covCd (See A_censoring and use_natural_course.)
classCd=	Categorical variables in covCd
covCn=	Baseline variables in model for denominator of weights , no treatment variables should be included in covCd . (See A_censoring, Anything listed here also needs to be listed in covMSMbv.)
classCn=	Categorical variables in covCn
eligible_cens =	Eligibility variable for Cens 1: yes, 0: no If 0 then pC_d and pC_n will be set to 1

Settings for second and third censoring models (Same as for cens).

Cens2= , covC2d= ,
classC2d= , covC2n= ,
classC2n= , eligible_cens2 =
Cens3=, covC3d= ,
classC3d= , covC3n= ,
classC3n=, eligible_cens3= ,

use_stabilized_wts = 1	Include the numerator model in the calculation of the weights
user_defined_weights = 0	Use user defined weights in analysis, skip calculate_weights macro

Data analysis settings.

class_ref = first	Method for coding reference level for binary dummy variables for class variables.This level will be represented when all binary variables are equal to 0. first = default coding is to use lowest level for all categorical variables. last = use largest level as reference level in coding the binary variables.
use_genmod = 0	When equal to 1, use PROC GENMOD for final weighted model for outcMSM when not running bootstraps or calculating the analytic variance.Can be used for obtaining an estimate of the robust variance. Otherwise use PROC LOGISTIC.
msm_var = 1	Calculate analytic estimate of variance
truncate_weights = 0	0 default: use all weights 1 truncates weights below and above 1 and 99 percentile 2 truncates weights below and above 2 and 98 percentile, etc

user_defined_limits =	User defined lower and upper limits for the weights. This must be a list with two numbers separated with a space: user_defined_limits = lower_limit upper_limit. this will only be used when truncate_weights = 1 and user_defined_limits is not missing. If only one entry is listed or the list is impty then the method will used the percentile values given in the truncate_weights option.

Bootstrap settings.

bootstrap= 0	Use bootstrap samples to estimate the variance of the parameters of the weighted models.
nboot= 200	Number of bootstrap samples to use.
bootstart = 0	Starting sample in current run of bootstrap samples (0 = original data).
bootend = 200	Last sample to use in current run of bootstrap samples (should be <= nboot.
bootlib = work	Libname for saving bootstrap results.
bootname = boot	Name of dataset for holding bootstrap results
bseed= 12345	Random number seed for boostrap

Computational settings

just_models = 0	Estimate the treatment and censoring weights only. Can be used for testing possible models.
override_warnings= 0	Override the warnings from the weight models. Normally, when the weight models do not converge any further analyses are not performed. Use this option when running bootstraps to continue when there are warnings.
logistic_options =	Possible options to include in the PROC LOGISTIC model statements. (e.g. possible increasing of maxiter).
inv_type = inv	When msm_var = 1, inv_type is the proc iml function that is used in calculating inverse of required matrices (possible choices are ginv = generalized inverse and inv = usual inverse) . Due to potential ill conditioned matrices one function may work when the other does not. Information concerning the determinant and estimated condtion number are supplied in the log file.

Output settings

no_listing = 0	Suppress all listing output useful when calling macro from a loop and do not desire the output.
print_boot= 0	Print out the individal model results for each bootstrap sample.
save_results= 1	Save bootstrap results into sas data sets.
save_weights= 0	Save the weights to a permanent data set. (The macro will keep the weights in a temporary data set called _weights.
libsurv= work	User defined libname for saving results, work directory will be default
results_name= _results	User defined name of file for saving results. (This is further explained in the examples.)

weights_name = _weights	User defined name of file to save calcuated weights in.
debug= 0	1 = keep intermediary data sets, 0 = delete intermediary data sets

Survival curves and cumulative incidence settings.

survival_curves = 0	Calculate the survival probabilities for two situations where A = 0 and A = 1 for all time points. (This can be generalized to other types of treatment histories as described in the examples.)
AMSM_type = 0	Default type of function of treatment will be AMSM = A 1 = A_m and $\frac{1}{m}\sum_{j=0}^{j=m-1} A_j$ 2 = Categorical variable, which is a function of time treated. For curves submacro user must list the times where the function changes. 3 = User defined function of treatment. For curves macro this will need to be coded by user.
AMSM_knots=	When amsm_type = 2, this lists the times where amsm changes levels.
use_natural_course = 0	Include a natural course analysis that removes the treatment weights and AMSM from weighted models (only used when survival_curves = 1). When use_natural_course=1 and survival_curves = 1, treatment variables are only available for use in the denominator models for censoring. In this case, all treatment variables must be included in A_censoring. This also forced all non natural course censoring models to contain the same function of treatment. When A_censoring is left blank, no treatment variables will be used in the natural course censoring models, but as described above, the treatment variable listed in A will be forced into all non natural course censoring models.
competing_risk =	An additional outcome for which a marginal structural model is fit. This is done assuming the same structural model and estimated weights as used for outcMSM. When using survival_curves = 1, a risk difference is also calculated at each time point. The user can select how this variable, riskdiff, is defined. The possible variables are km_nontreat, km_treat, km_nc for survival probabilities, for never treated, always treated and under the natural course . There are corresponding variables for the cumulative risk: ci_nontreat, ci_treat, and ci_nc. The variables for _nc are only included when use_natural_course = 1. When there is a competing risk, these variables are calculated using methods suggested in Gooley.
riskdiff1 = ci_treat	One variable from the list { km_nontreat, km_treat, km_nc, ci_nontreat, ci_treat, ci_nc }.
riskdiff0 = ci_nontreat	select from ci_nontreat ci_nc for calcluating ci_treated - riskdiff0
print_ci = 0	
time_start = 0	Time point to start calculating the survival probabilities.

time_end = 60	Final time point to use in the survival probabilities.
time_step = 1	Time step used in calculating survival probabilities. Will assume that if this is not 1 than the probabilies are constant between the various time point values.

2 Example of use

The sample dataset includes 1000 subjects with data measured at 6 timepoints (numbered 0 to 5). The outcome of interest is diabetes (dia). The fixed baseline covariates are age (baseage), hypertension (hbp_b), and bmi (bmi_b). There are also time varying versions of hypertension and bmi, along with two lagged values for bmi. We transformed the bmi values by setting bmi=log(bmi). Also included are two possible censoring variables: lost to followup (censlost) and a competing risk outcome (censdead). The treatment variable is A = I(bmi <= 25). The following call to %MSM describes how to fit a structural model with stabilized weights for treatment and a single censoring variable (censlost) using the analytic variance.

```
%msm(
      data = sampledata,
        id = id,
        time = time,
        outcMSM = dia,
        AMSM = A ,
        covMSMbh = time,
        classMSMbh = time,
        covMSMbv = baseage hbp_b bmi_b,
        msm_var = 1,

        cens = censlost,
        covCd = time baseage hbp_b bmi_b hbp,
        classCd = time,
        covCn = time baseage hbp_b bmi_b ,
        classCn = time,


        A = A ,
        covAd = time baseage hbp_b bmi_b hbp bmi_l1 bmi_l2,
        classAd = time,
        covAn = time baseage hbp_b bmi_b,
        classAn = time,
        A_censoring= A,

        save_results = 0,
        save_weights = 0
        ) ;
```

Using this call statement the macro will produce output for the four weight models (both numerator and denominator models for treatment, A, and censoring, censlost). Also, there are three PROC UNIVARIATE results for the components of the stabilized weights: wtA for treatment weights, wtC for censoring weights, and sw for the stabilized weights used in the weighted model for dia. In addition, if the option to truncate the weights is used, there are univariate results for the truncated weights. When using the default option of use_genmod = 0, the weighted model is performed using a pooled logistic model using PROC LOGISTIC. The output is after the analyses of the weights. The final table contains the standard errors for the parameters in the weighted model along with their 95% confidence bounds and the Wald Chi-square statistic and corrsponding p-value. From the above call the results should be:

Results for marginal structural model with stabilized weights for dia using analytic variance.

NAME	estimate	STD	LB	UB	Z	P_VALUE
Intercept	-3.64134	2.04522	-7.64997	0.36729	-1.78041	0.07501
A	-0.04313	0.31165	-0.65397	0.56770	-0.13840	0.88993
time1	0.67400	0.55647	-0.41668	1.76469	1.21121	0.22581
time2	0.67304	0.57056	-0.44525	1.79134	1.17962	0.23815
time3	0.65319	0.58411	-0.49167	1.79805	1.11826	0.26346
time4	-0.09758	0.72903	-1.52648	1.33132	-0.13385	0.89352
time5	1.14838	0.55752	0.05564	2.24113	2.05979	0.03942
baseage	-0.01961	0.02159	-0.06193	0.02271	-0.90830	0.36372
hbp_b	0.18946	0.31312	-0.42427	0.80318	0.60506	0.54514
bmi_b	-0.24879	0.58753	-1.40034	0.90277	-0.42344	0.67197

Since time was listed as in classMSMbh the 5 binary variables were created as time1,...,time5 where the suffix 1,..,5 denotes the corresponding value of time that is is representing (using the reference level of time = 0 from using class_ref = first).

3 %Bootstrap_results macro

Users may specify whether the variance is calculated analytically (as shown above) or via the nonparametric bootstrap method. The submacro %bootstrap_results displays the results for a single call to the %msm macro and for putting together the result files when running a bootstrap with more than a single call statement. In this section we describe the macro parameters and give an example where a set of 1000 samples is divided into 4 parts of length 250 and then combined to produce the final table of results. For simplicity, we will only describe the call statement when the survival probabilities are desired. An actual example is provided in the msm_samples.sas file using 25 bootstrap samples. The following macro parameters must match those used in calling the msm macro used to create the various parts.

The parameters of the %bootstrap_results macro are:

bootlib=	SAS library used to store the individual parts of the bootstrap run.
bootname=	Name (prefix only) of datasets used to store the bootstrap samples.
modeltype=	Table of results desired. sw = weighted model for outcMSM, nc = Natural course results, cr = competing risk reslts, surv = survival probabily results
numparts = 1	Number of parts to combine. This is the number of calls to the msm macro that was used to run the bootstrap,
samplestart = 0	Starting sample for each part to include. This is the value of bootstart used in each call. The first should be 0.
numboot = 200	Value of nboot used. This is the total number of bootstrap samples that were run.

time =	This is the time variable that was used in the call to the msm macro.
use_natural_course = 1	Same option as in the call to the msm macro.
outcMSM =	Same outcome variable used in the msm call.
competing_risk =	Same variable used in the msm call.
riskdiff1 = ci_treat	How the risk difference was defined in the msm call.
riskdiff0 = ci_nc	Variable used in the risk difference in the msm macro call.
print_ci = 0	Indicator for printing out the cumulative risk results

For an example call statement we will use the above 1000 samples split into 4 separate calls to the msm macro. All variables in the msm macro need to be the same except for the variables of bootstart and bootend. We assume that bootlib = mylib and bootname = bootstrap for holding the different parts. For simplicity we will assume that there is no competing risk and the natural course is not desired. In this case competing_risk is set to be blank and the variable use_natural_course = 0. We will assume that the risk differenct is for the survival probabilies of always treated verses never treated (riskdiff1 = km_treat and riskdiff0 = km_nontreat.). As in the supplied example we will set outcMSM = dia and time = time. In this case we can create four calls to the msm macro by using the following settings; (all will set nboot = 1000)

Call 1	bootstart = 0 bootend = 250
Call 2	bootstart = 251 bootend = 500
Call 3	bootstart = 501 bootend = 750
Call 4	bootstart = 751 bootend = 1000

For the survival probabilities, each call to the macro will create a data set that has the name bootstrap_surv_bootstart_bootend in the desired library. The first call will have a dataset called bootstrap_surv_0_250 to hold the 251 indidual survival probabilies. To put the data sets back together, we use the following call to bootstrap_results:

```
%bootstrap_results(
        bootlib = mylib,
        bootname = bootstrap,
        modeltype = surv,
        numparts = 4,
        samplestart = 0 251 501 751,
        numboot = 1000,
        time = time,
        outcMSM = dia,
        competing_risk = ,
        use_natural_course = 0,
        riskdiff1 = km_treat ,
        riskdiff0 = km_nontreat
        );
```

4 Notes and warnings

All variables used in the models can not have any missing values. The only exceptions are for the outcome variable, outcMSM, and any possible censoring variables.

The variables AMSM_knots and time_knots are used when survival_curves = 1. When AMSM_type = 2 then the variable AMSM_knots needs to include all possible boundary values of time that determine the categories of AMSM. For example, if AMSM = 0 for never treated and is 1 when treated for less than or equal to 3 periods and is 2 when treated for more than 3 periods, then the value of AMSM_knots should be set to 0 3, which includes the first bound of 0. In this case the correct levels will be reproduced in the curves sub macro. One check is to run the macro with the mprint option and make sure that the AMSM variables are being created correctly in the curves data set. In a similar way, when using categorical time values and running the survival curves option, the time points need to be created correctly. To do this, set contMSMbh = 0 and list all possible time values on time_knots, If there are 6 possible values from 0 to 5 then set time_knots = 0 1 2 3 4 5. Again, due to what is chosen for class_ref will determine how the binary variables are being defined and this option will also be used in the recreation of the binary variables in the curves data set.

5 License

6 References

- Robins JM. Marginal Structural Models. 1997 Proceedings of the Section on Bayesian Statistical Science. Alexandria, Virginia: American Statistical Association 1998;1–10.

- Robins JM. Marginal structural models versus structural nested models as tools for causal inference. In: Halloran E, Berry D, eds. Statistical Models in Epidemiology: The Environment and Clinical Trials. New York: Springer-Verlag, 1999;95–134.
- Hernan MA, Brumback B, Robins JM. Marginal structural models to estimate the causal effect of zidovudine on the survival of HIV-positive men. Epidemiology 2000;11:561-570.
- Gooley TA, Leisenring W, Crowley J, Storer BE. Estimation of failure probabilities in the presence of competing risks: New representations of old estimators. Statistics in Medicine 1999; 18; 695-706.

7 Appendix: Structure of the %MSM macro

The %MSM macro is structured into several modules:

- %MSM_SURV is the main driver. It inputs data, analysis and computing specifications.
- %SETDATA prepares the data to be analyzed by the macro %MSMCOX
- %CALCULATE_WEIGHTS, estimates model parameters for treatment and censoring densities that are used in the calculation of the stablized weights used to estimate the structural model.
- %MSMCOX estimates the parameters of a marginal structural Cox model based on weights estimated in %CALCULATE_WEIGHTS.
- %INITIALIZE_MSM_VAR, initializes macro variables that will be used in the calculation of the analytic variance under the non-doubly robust marginal structural Cox model
- %CALCULATE_MSM_VAR, calculates the analytic variance of the parameters of a non-doubly robust marginal structural Cox model
- %CURVES calculates estimated survival probabilities (risks) for the event of interest under two possible interventions for a binary treatment with possible competing risks and under the natural course.
- %BOOTSTRAP_RESULTS See Section 3.
- %NUMARGS, auxillary macro for finding number of entries in a macro list
- %CREATE_CLASS_LIST, create a list of class variables which appear in a list of variables
- %REMOVE_CLASS, remove entries in one list from a second list
- %CONVERT_CLASS, converts categorical variables into equivalent binary {0,1} variables. These variables are then used in the various models
- %REMOVE_EXTRANEOUS, removes entries in a class list if they are not contained in the corresponding covariate list, e.g. classAn and covAn

- **User supplied macros:**
- %EXTRA_VARIABLES_DEF Used to create user defined variables when calculating the survival probabilities and the cummulative incidence rates in the %CURVES macro.
- %AMSM_USER_DEF_MACRO0
- %AMSM_USER_DEF_MACRO1

  User defined functions of treatment under the never treat and always treat portion of survival probabilities. See Note in previous section about how these macros can be used to estimate risk under general static regimes.