LASSO user guide

2013-05-28 PsN 3.6.2

Reference and introduction

The lasso—a novel method for predictive covariate model building in nonlinear mixed effects models. Jakob Ribbing, Joakim Nyberg, Ola Caster, E. Niclas Jonsson. J Pharmacokinet Pharmacodyn (2007) 34:485–517

Covariate models for population pharmacokinetics and pharmacodynamics are often built with a stepwise covariate modelling procedure (SCM, available in PsN). When analysing a small dataset this method may produce a covariate model that suffers from selection bias and poor predictive performance. The lasso is a method suggested to remedy these problems. It may also be faster than SCM and provide a validation of the covariate model. In the lasso all covariates must be standardised to have zero mean and standard deviation one. Subsequently, the model containing all potential covariate–parameter relations is fitted with a restriction: the sum of the absolute covariate coefficients must be smaller than a value, t. The restriction will force some coefficients towards zero while the others are estimated with shrinkage. This means in practice that when fitting the model the covariate relations are tested for inclusion at the same time as the included relations are estimated. For a given SCM analysis, the model size depends on the P-value required for selection. In the lasso the model size instead depends on the value of t which can be estimated using cross-validation.

The lasso program does not support $PRIOR in the input model.

Example

lasso run2.mod -relations=CL:CLCR-2,WT-3,,V:WT-2 -seed=2011

Input and options

<u>lasso-specific input</u>

A model file is required on the command-line.

-relations=<list>	The relations option is required. The parameter-covariate pairs to test and the parameterizations to use. The parameterizations are: 1)categorical covariate (treated as nominal scale), 2)linear continuous covariate, 3)piece-wise linear “hockey-stick” for continuous covariate. The syntax is sensitive and must follow the form below. Note colons, commas. There must be no spaces in the list of relations. There are double commas before a new parameter. It is optional to input the breakpoint for the piece-wise linear relation (3), the number is then set after a second dash (-) as in WGT-3-45.2. Example (everything should be on the same line): -relations= CL:WGT-2,SEX-1,RACE-1,,V:WGT-3-45.2,,KA:WGT-3,APGR-2
-lst_file=file.lst	Name of the lst-file with estimates for the input model. Default is the name of the input model with a .mod extension replaced with .lst.

-groups=N	The number of validation groups in the cross-validation. Default 5, must be in the range 2-number of individuals in dataset. The larger the number the longer the cross-validation run-time.
-step_t=X	The steplength for t in the cross-validation. Default is 0.05. The step-length can be negative if start_t is larger than stop_t.
-start_t=X	The first t-value. Default is 0.
-stop_t=X	The last t-value. Default is 1.
-cutoff=X	The theta cutoff, if the absolute value of the estimated covariate theta is below cutoff then the theta will be fixed to zero. Default is 0.005.
-convergence='FIRSTMIN'	FIRSTMIN is the default. The convergence criterion. Alternatives are 'REACHMAX': Stop when program reaches max (stop_t), then best t-value is selected. 'FIRSTMIN': Stop when predicted ofv increases from previous value (or when stop_t is reached). 'HALT': Stop when program crashes or when one model cannot terminate.
-pred_ofv_start_t=X	Default not used. The total predicted ofv for the model with t=start_t. If the option is not set PsN will run the lasso model with t=start_t. If the value is given this will save time.
-stratify_on=variable	Default not used. If the option is set, PsN will try to preserve the relative proportions of the values of this variable when creating the cross-validation datasets. The variable must be in the dataset (not in an extra data file).

Some general PsN-options which are interesting in combination with lasso

For a complete list of common options see common_options_defaults_versions.pdf, or psn_options -h on the command-line.

The high dimensionality of a lasso model can make minimization more difficult. Therefore lasso has some special methods for handling minimization terminated due to rounding errors, and minimization terminated due to max number of evaluations exceeded.

Retries in lasso work slightly differently from retries in other PsN scripts. Before doing a retry in lasso, if minimization is terminated due to rounding errors then PsN will set thetas below cutoff to 0. Initial estimates will not be tweaked at the first retry, when thetas are cut for the first time. Instead the initial estimates will be set to the final estimates of the first try. If repeated retries are required then thetas will be cut again and then initial estimates will be tweaked. As with any PsN script the total number of retries is governed by the -retries option.

Picking the best retry results is done as in other PsN scripts, see details in common_options_defaults_versions_psn.pdf. It does not matter if some retries have thetas that are set to zero.

-directory=run1_lasso	The directory in which the script will run NONMEM can be named. The default name is “lasso_dirX” where X is increased by 1 each time you run the script.
-significant_digits_accept='number'	Default not used. Normally lasso tries new initial estimates unless 'MINIMZATION SUCCESSFUL' is found in the NONMEM output file. With the -significant_digits_accept, lasso will only

	rerun if the resulting significant digits is lower than the value specified with this option. Setting this option may reduce run times considerably. It may or may not affect the final results.
-maxevals=N	This option is important for lasso, since the difficult minimizations in the cross-validation may require many evaluations. In the template psn.conf maxevals is set to 50000. Will only work for classical estimation methods. NONMEM only allows 9999 function evaluations. PsN can expand this limit by adding an MSFO option to $ESTIMATION. Later when NONMEM hits the max number of function evaluations allowed by NONMEM (9999) PsN will remove initial estimates from the model-file and add $MSFI and restart NONMEM. This will be repeated until the number of function evaluations specified with option -maxevals has been reached. Note: PsN does not change the MAXEVALS setting in the model-file, therefore the number of evaluations set on the command-line may be exceeded before PsN does the check if the run should be restarted with msfi or not.
-retries=N	If minimization is not successful, PsN can do a retry with cut thetas (only lasso) and/or randomly tweaked inits. For extreme examples with lasso option retries may have to be as much as 20. On the other hand, -retries=0 is generally enough if rounding errors are acceptable (see above for the option significant_digits_accept)
-min_retries=N	Do not set this option with lasso; it may interfere with the handling of runs with rounding errors.
-seed=X	A seed for the random number generator can be specified. This makes the run reproducible. It acts as a starting point for the random number generator to produce the random number that will be used for creating the cross-validation data sets.
-threads=X	The number of parallel processes to start for the model runs on a parallel computer. Setting this value to something larger than the number of groups (cross-validation data sets) will have no effect.
-help	With -help lasso will print a longer help message.

Output

The optimal lasso model and the original model with the optimal covariate relations added are in the m1 subdirectory of the run directory. Located directly in the run directory: the file lasso.log contains the sequence of estimation and prediction ofv:s and the corresponding t-values. The files est_ofv.log and pred_ofv.log contain the estimation and prediction ofv in table format. The file coeff_table.log lists the estimated theta values for the covariates for each value of t in the cross-validation.

Reusing old lasso output

There are no special methods implemented to restart a crashed/stopped lasso run. Try starting with the

same command, with -directory set to existing run directory. Do not change option rerun from the default value (1).

Option -pred_ofv_start_t and -start_t can be set to values known from a previous run. This approach would require the use of a new run directory; i.e. do not set -dir to that of the previous run. The approach useful when one wants to take smaller t-steps in a particular region. Make sure the -seed is the same between the two runs, so that the cross-validation datasets are identical. If the optimal t-value is known set -start_t equal to -stop_t.

Example lasso model

With the command

lasso run1.mod -relations=CL:WGT-2,SEX-1,,V:WGT-2,APGR-3-5.5

where the input model file **run1.mod** is as follows,

```
$PROBLEM    PHENOBARB
$INPUT      ID TIME AMT WGT APGR DV SEX
$DATA       pheno_ch.dta IGNORE=@
$SUBROUTINE ADVAN1 TRANS2
$PK
    TVCL  = THETA(1)
    CL    = TVCL*EXP(ETA(1))
    TVV   = THETA(2)
    V     = TVV*EXP(ETA(2))
    S1    = V
$ERROR
    W     = THETA(3)
    Y     = F+W*EPS(1)
    IPRED = F         ;  individual-specific prediction
    IRES  = DV-IPRED  ;  individual-specific residual
    IWRES = IRES/W    ;  individual-specific weighted residual
$THETA  (0,0.0105)    ;CL
$THETA  (0,1.0500)    ;V
$THETA  (0,0.4)       ;W
$OMEGA  .4
      .25
$SIGMA  1 FIX
$ESTIMATION MAXEVAL=9999 SIGDIGITS=4 POSTHOC METHOD=1
```

the **initial lasso model** becomes (normalization constants depend on statistics computed from run1.mod data file)

```
$PROBLEM    PHENOBARB
$INPUT      ID TIME AMT WGT APGR DV SEX
$DATA      pheno_ch.dta IGNORE=@
$SUBROUTINE ADVAN1 TRANS2
$OMEGA  .4
```

```
 .25
$PK
;;; LASSO-BEGIN
   TVALUE  = THETA(9)
   ABSSUM = ABS(THETA(4))+ABS(THETA(5))+ABS(THETA(6))+ABS(THETA(7))
   ABSSUM = ABSSUM+ABS(THETA(8))

   RATIO = ABSSUM/TVALUE
   IF (RATIO.GT.5) EXIT 1 1
   FACTOR = EXP(1-RATIO)

   SEX0 = 0
   HAPGR = 0
   IF (SEX.EQ.0) SEX0=1
   IF (APGR.GT.5.50000) HAPGR = APGR-5.50000

   CLWGT = THETA(4)*(WGT-1.52542)/0.70456*FACTOR
   CLSEX0 = THETA(5)*(SEX0-0.49153)/0.50422*FACTOR
   VAPGR = THETA(6)*(APGR-6.42373)/2.23764*FACTOR
   VHAPGR = THETA(7)*(HAPGR-1.49153)/1.33420*FACTOR
   VWGT = THETA(8)*(WGT-1.52542)/0.70456*FACTOR

   VCOV = (VAPGR+1)*(VHAPGR+1)*(VWGT+1)
   CLCOV = (CLWGT+1)*(CLSEX0+1)
;;; LASSO-END

    TVCL  = THETA(1)
    TVCL = TVCL*CLCOV
    CL    = TVCL*EXP(ETA(1))
    TVV   = THETA(2)
    TVV = TVV*VCOV
    V     = TVV*EXP(ETA(2))
    S1    = V
$ERROR
    W     = THETA(3)
    Y     = F+W*EPS(1)
    IPRED = F         ;  individual-specific prediction
    IRES  = DV-IPRED   ;  individual-specific residual
    IWRES = IRES/W     ;  individual-specific weighted residual

$THETA  (0,0.0105) ; CL
$THETA  (0,1.0500) ; V
$THETA  (0,0.4) ; W
$THETA  (-0.33962,0.000100,0.76134) ; TH4 CLWGT
$THETA  (-0.99163,0.000100,1.02583) ; TH5 CLSEX0
$THETA  (-0.62569,0.000100,0.41256) ; TH6 VAPGR
```

```
$THETA  (-0.44348,0.000100,0.89452) ; TH7 VHAPGR
$THETA  (-0.33962,0.000100,0.76134) ; TH8 VWGT
$THETA  (-1000000,0.000000) FIX ; TH9 T-VALUE
$SIGMA  1  FIX
$ESTIMATION MAXEVAL=9999 SIGDIGITS=4 POSTHOC METHOD=1
```

Subsequent models tested during cross-validation have other values of T-VALUE.

Step-by-step overview of algorithm

1. Input checking.
2. Compute covariate statistics.
3. Create the lasso model by adding all covariate relations to input model. Normalize parameters using covariate statistics. See details in lasso reference article.
4. Create cross-validation data sets, possibly using stratification. Divide data into N groups. Create N estimation datasets where estimation dataset j is the total dataset minus group j. There are N prediction datasets where prediction dataset j is identical to group j.
5. Set t=start_t.
6. Run an xv step with lasso model unless option pred_ofv_start_t is given. An xv step consists of N estimations, one for each estimation dataset, and N predictions, one for each prediction dataset. The prediction j is a MAXEVAL=0 run with initial estimates set to the final estimates from estimation run j. (*Note, the default starting position is start_t=0, where cross validation of the base model is performed*)
7. Set t=t+step_t and run an xv step (see step 6).
8. Check convergence. If convergence is not reached, goto 7. If converged continue to 8.
9. Run the lasso model with optimal t on the whole (original) dataset.
10. Add the covariate relations and parameters from the optimal lasso model run (only relations with thetas larger than cutoff) to the input model, fix the selected covariate thetas as the lasso restriction has been removed and run the input model on the whole dataset.