CMAT Newsletter: June 2004

Wolfgang M. Hartmann

June 2004

1 General Remarks

This year I'm most of the time working for SAS Institute in Cary NC. Therefore, my output for CMAT is not as large as usual. In addition I had some work to organize our minisymposium at the PARA'04 conference at the TU Denmark in Lyngby Denmark. My talk there deals with feature selection methods for the analysis of microarray data. That is the reason that most of the new features are concerned with this problem.

2 Modifications of Features

2.1 Fixed Bugs

2.2 Extensions in `sem()` Function

With the help of a good friend I was able to enhance the algorithms for the Satorra-Bentler nonnormal modifications of the `sem` and `factor` functions so that now larger problems can be solved. For estimation methods other than WLS, the Satorra-Bentler modifications (goodness-of-fit, approximate standard errors, adjusted residuals) now no longer need core memory of size $nn2 * (nn2 + 1)/2$, where $nn2 = n * (n + 1)/2$ and n is the number of variables. The largest memory allocation is now a matrix of size $nn2 * np$, where np is the number of parameters. Most computer hardware will now permit to compute the Satorra-Bentler modifications for more than onehundred variables and a few hundred parameters. For example, for $n = 200$ variables and $np = 600$ parameters this matrix needs *only* about 92 Mbyte RAM.

Note, that the `sem` and `factor` functions now permit both forms of S-B modified χ^2 values, approximate standard errors and adjusted residuals for all estimation methods: ULS, GLS, ML, and DWLS. (For the WLS estimation method this is a natural property.)

More work must be done: as in LISREL, some other goodness-of-fit measures like the RMSEA should be based on the S-B corrected χ^2 value which will add a few lines in the table of GOF values. Also, the `noharm()` function will soon be extended with similar modifications, making larger applications possible.

2.3 Extensions in `pls()` Function

1. The randomization (or permutation) test by H. van der Voet (1994) was added. A few new options had to be added.

2. The options argument was changed from vector to two column matrix form. See below for the new form of specifying the options.

Option Name	`Second Column`	Meaning
`"block"`	int	fold number for block cross validation
`"cloo"`		compute Loo (leave-one-out) estimates
`"fold"`	int	fold number for block, split, and random cross validation (def=10)
`"ftol"`	real	termination criterion for NIPALS (def=1.e-6)
`"maxit"`	int	max. number iterations of NIPALS (def=1000)
`"meth"`		method of estimation (def=PLS)
	"pcr"	principal components regression
	"pls"	partial least squares
`"nopr"`		no printed output
`"ppred"`		print predicted values
`"ppar"`		print vector of parameter estimates
`"print"`	int	amount of general printed output (default is 3)
		=0: no output, =1: print only summary tables, =2: component-wise tables added, =3: parameters added, =4: predicted values)
`"pval"`	real	threshold probability for permutation test (def=0.1)
`"random"`	int	fold number for random cross validation
`"samsiz"`	int	sample size for permutation test
`"seed"`	int	seed for random generator; default is time of day; affects the results of random cross validation: affects results of permutation test
`"stat"`		statistics for permutation test
	"t2"	Hotellings T^2 (def.)
	"press"	predicted residual-sum-of-squares
`"split"`	int	fold number for split cross validation
`"tech"`	"simpls"	SIMPLS
	"krnpls"	Kernel PLS
	"nipals"	NIPALS

Note, this implementation of `pls` , similar to that in R, is different from many others, like that of SAS Institute, that it only centers the data and does not scale. For many applications it will therefore result in models which have better prediction than methods which scale the data.

The Biological Activity Example in SAS/STAT

The PROC PLS description in the SAS/STAT manual contains the following data set containing of 16 observations and 30 variables from which the first 27 are selected as predictors and the last three are selected for a multivariate response.

```
plsx = [ 2766 2610 3306 3630 3600 3438 3213 3051 2907 2844 2796
         2787 2760 2754 2670 2520 2310 2100 1917 1755 1602 1467
         1353 1260 1167 1101 1017         3.0110  0.0000   0.00,
.....................................................................
         4017 4725 6090 6570 6354 5895 5346 4911 4611 4422 4314
         4287 4224 4110 3915 3600 3240 2913 2598 2325 2088 1917
         1734 1587 1452 1356 1257         3.1620  0.7012  60.00 ];
   nr = nrow(plsx); nc = ncol(plsx);
   print "nr,nc=", nr,nc;
   cnam = [ "v1":"v27"  "ls"  "ha"  "dt" ];
   rnam = [ "EM1:EM16" ];
   plsex = cname(plsx,cnam);  /* plsex = rname(plsx,rnam); */
   yind = [ 28:30 ]; xind = [ 1:27 ];
   cmp = [ 1:15 ];
```

Selecting the Number of Factors by Split Cross Validation:

```
    optn = [ "prin"         1 ,
             "meth"     "pls" ,
             "tech"  "krnpls" ,
             "split"       10 ,
             "stat"      "t2" ];
    < rms,bmat,yprd > = pls(plsex,yind,xind,cmp,optn);


                    ****************************
                    Kernel Partial Least Squares
                    ****************************

    X Dimension . . . . . .       27
    Y Dimension . . . . . .        3
    Nobs Training Data. . . .     16
    Number Analyses . . . . .     14
    Maximum Number Factors. . .   14
    Cross Validation. . . . . .  Split
    Cross Validation Fold . . .   10
    Cross Validation Size . . .    2


                     -------------------------
                     Analysis of Training Data
                     -------------------------

                      Summary Table: Training
                      -----------------------
```

Factor	X_VarExp	Y_VarExp	RMS	R2
1	0.97991572	0.27197375	0.43447075	0.87984913
2	0.99717018	0.87451641	0.17739128	0.13082423
3	0.99863012	0.91196597	32.0013435	0.27129784
4	0.99892476	0.94238582	0.23583256	0.96459918
5	0.99921147	0.96714482	0.17447785	0.15914003
6	0.99994051	0.96945055	13.2840035	0.87443413
7	0.99996324	0.98422567	0.23575956	0.96462109
8	0.99997420	0.99339110	0.17036408	0.19832354
9	0.99998113	0.99787628	11.1254682	0.91192549
10	0.99998534	0.99946052	0.20435763	0.97341801
11	0.99998878	0.99992021	0.14775380	0.39699572
12	0.99999570	0.99994962	8.99985768	0.94236518
13	0.99999737	0.99998960	0.13101926	0.98907362
14	0.99999904	0.99999963	0.10927539	0.67017176

Summary Table: Variance Explained

Factor	Predictors	ls	ha	dt
1	0.97991572	0.87984913	0.13082423	0.27129784
2	0.99717018	0.96459918	0.15914003	0.87443413
3	0.99863012	0.96462109	0.19832354	0.91192549
4	0.99892476	0.97341801	0.39699572	0.94236518
5	0.99921147	0.98907362	0.67017176	0.96712796
6	0.99994051	0.99815629	0.96397010	0.96941860
7	0.99996324	0.99869773	0.97220579	0.98420981
8	0.99997420	0.99914291	0.97921250	0.99338503
9	0.99998113	0.99914748	0.98139841	0.99787529
10	0.99998534	0.99927562	0.98508684	0.99946110
11	0.99998878	0.99929860	0.98577669	0.99992127
12	0.99999570	0.99948066	0.98773372	0.99995046
13	0.99999737	0.99993147	0.99031475	0.99998992
14	0.99999904	0.99993189	0.99766310	0.99999977

Summary Table: RMS

Factor	ls	ha	dt
1	0.43447075	0.17739128	32.0013435
2	0.23583256	0.17447785	13.2840035
3	0.23575956	0.17036408	11.1254682
4	0.20435763	0.14775380	8.99985768
5	0.13101926	0.10927539	6.79683181

6	0.05381997	0.03611686	6.55574249
7	0.04523217	0.03172164	4.71071598
8	0.03669527	0.02743340	3.04899730
9	0.03659743	0.02595096	1.72799907
10	0.03373494	0.02323611	0.87025759
11	0.03319563	0.02269232	0.33264155
12	0.02856413	0.02107341	0.26384986
13	0.01037601	0.01872552	0.11902617
14	0.01034436	0.00919812	0.01798036

Summary Table: Rsquared

Factor	ls	ha	dt
1	0.87984913	0.13082423	0.27129784
2	0.96459918	0.15914003	0.87443413
3	0.96462109	0.19832354	0.91192549
4	0.97341801	0.39699572	0.94236518
5	0.98907362	0.67017176	0.96712796
6	0.99815629	0.96397010	0.96941860
7	0.99869773	0.97220579	0.98420981
8	0.99914291	0.97921250	0.99338503
9	0.99914748	0.98139841	0.99787529
10	0.99927562	0.98508684	0.99946110
11	0.99929860	0.98577669	0.99992127
12	0.99948066	0.98773372	0.99995046
13	0.99993147	0.99031475	0.99998992
14	0.99993189	0.99766310	0.99999977

****
Validation of Number of Extracted Factors
****

Nfac	R_M_PRESS	T**2	Prob>T**2
1	21.2609410	0.43165532	0.01500000
2	10.4871916	0.15124027	0.61700000
3	9.92398812	.	.
4	16.2703024	0.23070548	0.23200000
5	16.6296045	0.21946175	0.34100000
6	12.2703502	0.47403961	0.00800000
7	11.5056415	0.49475981	0.00300000
8	14.0197225	0.46933272	0.01300000
9	12.2440887	0.53718999	0.00400000
10	11.9636972	0.57985503	0.00400000

```
11  11.3513559  0.61109551  0.00200000
12  11.5877833  0.61315804  0.00100000
13  11.5910951  0.61062540  0.00100000
14  11.5999137  0.60695255  0.00200000
```

```
Minimum Root Mean PRESS 9.92399 for 3 Factors
Smallest Number of Factors with p > 0.1 is 2
```

```
-----------------------------------
K=10-fold Cross Validation (Split)
-----------------------------------
```

```
Best Number of Factors wrt. Cross Validation: 14
```

```
Summary Table: RMS
------------------
```

Factor	ls	ha	dt
1	0.46480237	0.20170001	36.8215441
2	0.32594438	0.24182467	18.1598140
3	0.34277827	0.26010919	17.1834649
4	0.31763472	0.23314614	28.1782358
5	0.32810617	0.20323548	28.8007341
6	0.18245596	0.10881638	21.2518082
7	0.17426173	0.11059122	19.9272869
8	0.14133616	0.11924670	24.2821675
9	0.11732582	0.14077270	21.2065920
10	0.09701342	0.14126572	20.7210228
11	0.08560470	0.13478843	19.6604768
12	0.09596884	0.15739319	20.0697828
13	0.11186080	0.14231473	20.0755496
14	0.10727714	0.14280162	20.0908461

```
Summary Table: RMS_Stdev
------------------------
```

Factor	ls	ha	dt
1	0.12001137	0.05207841	9.50661570
2	0.08402525	0.06238388	4.68731380
3	0.08836285	0.06705892	4.43032581
4	0.08122501	0.05960069	7.06518085
5	0.08379805	0.05230871	7.16136542

```
 6  0.04479110  0.02789555  5.31494541
 7  0.04295632  0.02853957  5.09615479
 8  0.03513222  0.03054172  6.12084765
 9  0.02988138  0.03617580  5.29768400
10  0.02477027  0.03611721  5.23108664
11  0.02202603  0.03445286  4.96519145
12  0.02476166  0.04063162  5.07009773
13  0.02887575  0.03673857  5.07184929
14  0.02767053  0.03684882  5.07580596


     Summary Table: Rsquared
     -----------------------

Factor          ls          ha          dt
     1  0.86251797  0.00392416  0.09095948
     2  0.93340813  0.04029134  0.76828855
     3  0.92560901  0.03916215  0.80470080
     4  0.94397962  0.00949961  0.74698549
     5  0.93754516  0.21205980  0.78291056
     6  0.98291640  0.68015273  0.84637921
     7  0.98400547  0.67098435  0.84165007
     8  0.98984183  0.62005073  0.78802564
     9  0.99326819  0.49288684  0.81758896
    10  0.99499108  0.50248584  0.81182429
    11  0.99604077  0.53235365  0.82000105
    12  0.99468012  0.38306767  0.81625296
    13  0.99246562  0.47650942  0.81603401
    14  0.99294316  0.47785323  0.81576806


      Time for Training: 0
    Time for Validation: 0
         Total Time: 0
```

2.4 Extensions in `svm()` Function

Some new methods for feature selection are now available with the `svm()` function.

1. `"task" "clas"` or `"task" "regr"` and `"meth" "lssvm"`:
   The LS (least squares) support vector machine algorithm was added, for both, C classification and L2 regression. For classification, this method is very similar to the PSVM algorithm by Fung & Mangasarian. For regression the LS SVM method is very similar to *ridge regression*. It only

requires to solve a (ridged, i.e. positive definite) large $N \times N$ linear system. Two solution methods are implemented:

(a) The direct Cholesky method (dense or sparse) requires the coefficient matrix in memory and is suitable for only small N.

(b) The iterative conjugate gradient algorithm does not require the coefficient matrix incore and computes the matrix-vector product by creating the kernel matrix entries "on the fly". This approach, however, requires the $N \times n$ data matrix incore.

Both, LS SVM for classification and for regression were developed and researched by the Suykens and Vandevalle team of the Kath. University of Leeuven. The method is (like PSVM) very fast but creates nonsparse estimates.

2. `"task" "sfsr"` and `"meth" "fqp"`, `"meth" "asvm"` `"meth" "lsvm"`, or `"meth" "psvm"`:
   The stepwise (greedy) variable selection method by Guyon et.al. (2002). In each step a number of features with smallest absolute weights is dropped creating so a series of nested variable subsets. The amount of variables dropped at each step can be controlled by the `"vardrop"` option and the smalest number of subsets which is also the termination criterion of the algorithm can be specified by the `"nvarmin"` option. This algorithm is performed for the `"sfsr"` task and a number of methods as `"fqp"`, `"asvm"`, `"lsvm"`, and `"psvm"`. Note, during the first stage of this method always `"nsvm"` is used as long as $N \leq n$, then at the second stage when $N > n$ the stepwise algorithm switches to the user specified method.

3. `"task" "nure"` and `"meth" "flp"`:
   The FLP method can be used for solving the ν regression problem with linear kernel for $N \ll n$. This algorithm is described in Bi et. al. (2002). Two algorithms can be used currently for solving the very large and dense LP problem, *pcx* by Czyzyk, Mehrotra, Wagner, and Wright (1997) and the continuation algorithm by Madsen, Nielsen, and Pinar (1996). The interior point algorithm *pcx* seems to be more stable numerically and is therefore the default choice, but the continuation algorithm can be much faster for well behaved problems. The sparsity of the linear plane solution depends on the specification of the hyper parameter ν. Parameter tuning with grid search is recommended. This algorithm is performed for the `"nure"` task and the `"flp"` method.

An additional kernel definition was added to the `svm` function: the *compactly supported Gaussian RBF* kernel

$$K(u,v) = \max(0, (1 - \frac{\parallel u - v \parallel}{\theta})^{\kappa}) exp(-\frac{\parallel u - v \parallel}{\sigma^2})$$

as proposed by Hamers, Suykens, & DeMoor (2002) and Genton (2001). For $\parallel u - v \parallel > \theta$ this type of kernel will introduce some amount of sparsity in the

kernel matrix $\mathbf{K}$ and $\kappa \geq (n+1)/2$ is an odd integer close to n. The default for θ is $\theta = 3 * \sigma$ as recommended in Hamers et.al. (2001). The kernel function can be specified as `"kern" "rbfcs"` and the factor of σ (which is equal 3 by default) using the `"sigfac"` option. Larger values of the factor increase the degreee of sparsity.

1. First example for feature selection with ν regression:

   /*— FQP: nvarmin= 3, 5, 7, 10 ————*/

```
optn = [ "print"           3 ,
         "pplan"             ,
         "task"       "sfsc" ,
         "meth"        "fqp" ,
         "nvarmin"         3 ,
         "vardrop"        .2 ,
         "fold"           10 ,
         "c"              1. ,
         "kern"       "line" ];
alfa = svm(tnir,modl,optn,class);
```

```
*******************************************
Stepwise Variable Selection (Ntrn=21 Nvar=268)
   Size: Trainig Data=10 Validation Data=11
*******************************************
```

Iter	NDrop	WgtDrop	NRem	WgtRemain	MiscTrn
1	0	.	268	11.546638	.
2	54	0.3202434	214	11.334699	.
3	43	0.6239116	171	7.5385469	.
4	34	0.3093094	137	10.523981	.
5	27	1.2406915	110	8.0957709	.
6	22	0.7713951	88	5.4381890	.
7	18	0.5859408	70	5.9357352	.
8	14	0.6946530	56	4.1798109	.
9	11	0.5979633	45	3.6850981	.
10	9	0.5665638	36	3.5387044	.
11	7	0.5319841	29	2.4866877	.

Iter	NDrop	WgtDrop	NRem	WgtRemain	MiscVal
12	6	0.3723136	23	2.3945302	.
13	5	0.4540156	18	2.0681687	.
14	4	0.3959728	14	1.7774829	.
15	3	0.3275218	11	0.5769029	.

```
16      1  0.0398765     10  1.3806106        .
17      2  0.2123163      8  2.2412667        1
18      2  0.3114333      6  3.9698555        2
19      1  0.4859327      5  2.2713750        1
20      1  0.2374750      4  3.7636927        1
21      1  0.6223037      3  1.6570000        1


                Classification Table
                --------------------

                         |    Predicted
                Observed |        0       1
                ---------|-----------------
                       0 |        9       2
                       1 |        2       8


          Nonzeros of Separating Plane (w*x = 6.00227)
          --------------------------------------------

                    1  53  X53   0.049347450
                    2  71  X71  -2.071485192
                    3 129  X129 -0.750701791


             Largest 3 Plane Coefficients (Sorted)
             *************************************

                    1  71  X71  -2.071485192
                    2 129  X129 -0.750701791
                    3  53  X53   0.049347450
             Sparsity: 3 Nonzeros 265 Zeros (C=1)


optn = [ "print"           3 ,
         "pplan"             ,
         "task"       "sfsc" ,
         "meth"        "fqp" ,
         "nvarmin"        5 ,
         "vardrop"       .2 ,
         "fold"          10 ,
         "c"             1. ,
         "kern"       "line" ];
alfa = svm(tnir,modl,optn,class);
```

```
*********************************************
Stepwise Variable Selection (Ntrn=21 Nvar=268)
   Size: Trainig Data=10 Validation Data=11
*********************************************
```

Iter	NDrop	WgtDrop	NRem	WgtRemain	MiscTrn
1	0	.	268	11.546638	.
2	54	0.3202434	214	11.334699	.
3	43	0.6239116	171	7.5385469	.
4	34	0.3093094	137	10.523981	.
5	27	1.2406915	110	8.0957709	.
6	22	0.7713951	88	5.4381890	.
7	18	0.5859408	70	5.9357352	.
8	14	0.6946530	56	4.1798109	.
9	11	0.5979633	45	3.6850981	.
10	9	0.5665638	36	3.5387044	.
11	7	0.5319841	29	2.4866877	.
12	6	0.3723136	23	2.3945302	.
13	5	0.4540156	18	2.0681687	.
14	4	0.3959728	14	1.7774829	.
15	3	0.3275218	11	0.5769029	.
16	1	0.0398765	10	1.3806106	.

Iter	NDrop	WgtDrop	NRem	WgtRemain	MiscVal
17	2	0.2123163	8	2.2412667	1
18	2	0.3114333	6	3.9698555	2
19	1	0.4859327	5	2.2713750	1

Classification Table

Observed	Predicted 0	Predicted 1
0	9	2
1	1	9

Nonzeros of Separating Plane (w*x = 1.59952)

1	51	X51	0.911170405
2	52	X52	0.513434877
3	53	X53	-0.027874358

```
                4  71  X71  -1.857537964
                5 129  X129 -0.559404324


            Largest 5 Plane Coefficients (Sorted)
            *************************************

                1  71  X71  -1.857537964
                2  51  X51   0.911170405
                3 129  X129 -0.559404324
                4  52  X52   0.513434877
                5  53  X53  -0.027874358
            Sparsity: 5 Nonzeros 263 Zeros (C=1)


optn = [ "print"           3 ,
         "pplan"             ,
         "task"       "sfsc" ,
         "meth"        "fqp" ,
         "nvarmin"         7 ,
         "vardrop"        .2 ,
         "fold"           10 ,
         "c"              1. ,
         "kern"       "line" ];
alfa = svm(tnir,modl,optn,class);


                  Classification Table
                  --------------------

                         |    Predicted
                Observed |        0       1
                ---------|-----------------
                       0 |       10       1
                       1 |        1       9


          Nonzeros of Separating Plane (w*x = 0.172672)
          ---------------------------------------------

                1  50  X50   0.946425671
                2  51  X51   0.897577640
                3  52  X52   0.497195611
                4  53  X53  -0.045991117
                5  54  X54  -0.333011904
```

```
                6  71  X71  -1.807344497
                7 129  X129 -0.514524645


            Largest 7 Plane Coefficients (Sorted)
            *************************************

                1  71  X71  -1.807344497
                2  50  X50   0.946425671
                3  51  X51   0.897577640
                4 129  X129 -0.514524645
                5  52  X52   0.497195611
                6  54  X54  -0.333011904
                7  53  X53  -0.045991117
            Sparsity: 7 Nonzeros 261 Zeros (C=1)


optn = [ "print"           3 ,
         "pplan"             ,
         "task"       "sfsc" ,
         "meth"        "fqp" ,
         "nvarmin"        10 ,
         "vardrop"        .2 ,
         "fold"           10 ,
         "c"              1. ,
         "kern"       "line" ];
alfa = svm(tnir,modl,optn,class);


                  Classification Table
                   --------------------

                           |    Predicted
                  Observed |        0       1
                  ---------|-----------------
                         0 |       10       1
                         1 |        0      10


          Nonzeros of Separating Plane (w*x = 6.30199)
          --------------------------------------------

                1  50  X50   0.685826697
                2  51  X51   0.617804429
                3  52  X52   0.221769616
```

```
 4  53  X53  -0.289055681
 5  54  X54  -0.533746566
 6  55  X55  -0.447167211
 7  71  X71  -0.945796697
 8  72  X72  -1.154633740
 9  73  X73  -1.213455401
10 129  X129  0.138733845


Largest 10 Plane Coefficients (Sorted)
**************************************

 1  73  X73  -1.213455401
 2  72  X72  -1.154633740
 3  71  X71  -0.945796697
 4  50  X50   0.685826697
 5  51  X51   0.617804429
 6  54  X54  -0.533746566
 7  55  X55  -0.447167211
 8  53  X53  -0.289055681
 9  52  X52   0.221769616
10 129  X129  0.138733845
Sparsity: 10 Nonzeros 258 Zeros (C=1)
```

2. Second example for feature selection with ν regression:

   A modified form of the algorithm of Bi et al. (2002) is performed on the Caco2 data with nobs= 27 and nvar = 714:

```
/* Caco: nobs= 27, nvar= 714 (Bi & Bennett) */
fo10 = " %16s %16s %16s %16s %16s %16s %16s %16s %16s %16s";
form = fo10;
for (j = 2; j <= 71; j++) form = strcat(form,fo10);
fo4 = " %16s %16s %16s %16s";
form = strcat(form,fo4);


fid = fopen("..\\tdata\\caco_nam.txt","r");
/* sscanf() is faster when nr and nc are specified: */
c_nam = fscanf(fid,form,.,1);
/* print c_nam; */


/*--- read data: 715 cols, first column is obs number ---*/
fid = fopen("..\\tdata\\caco.dat","r");
```

```
form = fo1 = " %g %g %g %g %g %g %g %g %g %g";
for (j = 2; j <= 71; j++) form = strcat(form,fo1);
fo5 = " %g %g %g %g %g";
form = strcat(form,fo5);
c_dat = fscanf(fid,form,27,.);
nr = nrow(c_dat); nc = ncol(c_dat);
print "CACO2: Observations of c_dat.dat:",nr;
print "CACO2: Columns of c_dat.dat:",nc;
caco = c_dat[,2:715]; /* cut out col 1 */


/* NUREG: meth=FSM: use PCX */
c = [ 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 12. ];
nu = [ .2 .5 .8 ];
tun_g = c |> nu ;
modl = "714 = 1 : 713";
optn = [ "print"           2 ,
         "pplan"             ,
         "task"      "nureg" ,
         "meth"        "fsm" ,
         "kern"       "line" ];
alfa = svm(caco,modl,optn,.,tun_g);
```

```
                    *****************
                     Model Information
                     *****************

               Number Valid Observations   27
               Response Variable        Y[714]
               N Independend Variables    713
               Support Vector NU Regression
               Model with 2 Linear Constraints
               Estimation Method          FSM
               Kernel Function         Linear
               NU Insensitivity   0.037037037
               Use Unscaled Predictor

     Traindata fit incore: Mem need: 0.15 Mem spec: 2 Mb
       Memory needed for Kernel matrix: 0.00288391 Mb


    ****************************************************
    **********- Evaluation for Best C=10 Nu=0.2-**********
    ****************************************************
```

```
Regularization Parameter C . . . . . 10
Parameter NU . . . . . 0.2
Kernel Function. . . . . Linear
Norm of Longest Vector . . . . . 49.1966
Mean Squared Error (MSE, Training Data). . . . . 3.84216e-018
Mean Absolute Error (MAE, Training Data) . . . . . 1.6817e-009
Average Loss (Training Data) . . . . . 1.6817e-009
Average Positive Loss (Training Data). . . 1.62164e-009 (N=14)
Average Negative Loss (Training Data). . . 1.74638e-009 (N=13)
Number Fast Runs through TrainData . . . . . 135
Number Slow Runs through TrainData . . . . . 137
Total Number of Kernel Calls . . . . . 2107
Time for Optimization. . . . . 550
Total Processing Time. . . . . 550
```

Grid Search Results: L1 Error (Training)

****

Dense Matrix (11 by 3)

	Nu0.2	Nu0.5	Nu0.8
c1	0.9629629	0.9629629	0.9629629
c2	0.1572939	0.1572939	0.1572939
c3	0.1050956	0.1050956	0.1050956
c4	0.0471402	0.0471402	0.0471402
c5	0.0255726	0.0255726	0.0255726
c6	0.0015505	0.0015505	0.0015505
c7	0.0012633	0.0012633	0.0012633
c8	0.0010429	0.0010429	0.0010429
c9	6.32e-014	6.32e-014	6.32e-014
c10	3.84e-018	3.84e-018	3.84e-018
c12	2.26e-014	2.26e-014	2.26e-014

Grid Search Results: L2 Error (Training)

****

Dense Matrix (11 by 3)

	Nu0.2	Nu0.5	Nu0.8
c1	26.000000	26.000000	26.000000
c2	4.2469345	4.2469345	4.2469345
c3	2.8375807	2.8375807	2.8375807

c4	1.2727847	1.2727847	1.2727847
c5	0.6904596	0.6904596	0.6904596
c6	0.0418639	0.0418639	0.0418639
c7	0.0341079	0.0341079	0.0341079
c8	0.0281582	0.0281582	0.0281582
c9	1.71e-012	1.71e-012	1.71e-012
c10	1.04e-016	1.04e-016	1.04e-016
c12	6.09e-013	6.09e-013	6.09e-013

FSM Grid Search: N Nonzero Coefficients

****

Dense Matrix (11 by 3)

	Nu0.2	Nu0.5	Nu0.8
c1	0	0	0
c2	4	4	4
c3	13	13	13
c4	18	18	18
c5	473	473	473
c6	84	84	84
c7	26	26	26
c8	27	27	27
c9	358	358	358
c10	26	26	26
c12	613	613	613

Optimization Criterion . . . . . .

Geometric Margin . . . . . 2.98598 (|w|^2= 0.448629)

Number Support Vectors . . . . . 27 (100.00 %)

Number Support Vectors on Margin 0

Bias 7.40797e-009

Radius of Sphere Around SV 49.1864

Estimated VCdim of Classifier. 1086.37

Linear Kernel Constant (Fit) 0

Linear Kernel Constant (PCE) . . . . . .

Nonzeros of Separating Plane (w*x = 0)

1	75	X75	0.132796352
2	253	X253	-0.051550705

3	270	X270	-0.014653077
4	281	X281	-0.025364047
5	326	X326	0.004793089
6	332	X332	0.046685150
7	382	X382	-0.038856622
8	488	X488	0.074328501
9	489	X489	0.020548493
10	557	X557	-0.026545510
11	620	X620	0.019289482
12	622	X622	0.012542633
13	647	X647	0.128653347
14	652	X652	0.025513393
15	662	X662	0.046067449
16	667	X667	0.036495523
17	679	X679	0.601082078
18	681	X681	-0.056335728
19	682	X682	-0.000564896
20	683	X683	0.017396180
21	684	X684	0.105761385
22	686	X686	0.000555438
23	687	X687	0.072067846
24	689	X689	-0.114096580
25	694	X694	-0.039462570
26	699	X699	0.014048546

Largest 26 Plane Coefficients (Sorted)

****

1	679	X679	0.601082078
2	75	X75	0.132796352
3	647	X647	0.128653347
4	689	X689	-0.114096580
5	684	X684	0.105761385
6	488	X488	0.074328501
7	687	X687	0.072067846
8	681	X681	-0.056335728
9	253	X253	-0.051550705
10	332	X332	0.046685150
11	662	X662	0.046067449
12	694	X694	-0.039462570
13	382	X382	-0.038856622
14	667	X667	0.036495523
15	557	X557	-0.026545510
16	652	X652	0.025513393
17	281	X281	-0.025364047

```
        18 489  X489  0.020548493
        19 620  X620  0.019289482
        20 683  X683  0.017396180
        21 270  X270 -0.014653077
        22 699  X699  0.014048546
        23 622  X622  0.012542633
        24 326  X326  0.004793089
        25 682  X682 -0.000564896
        26 686  X686  0.000555438
Sparsity: 26 Nonzeros 687 Zeros (C=10)


  Total Number of Kernel Calls: 2107
      Time for Optimization: 550
      Total Processing Time: 550
```

Similar results are obtained when using the continuation algorithm by Madsen, Nielsen, & Pinar (1996).

```
*****************************************************
**********- Evaluation for Best C=10 Nu=0.2-**********
*****************************************************

Regularization Parameter C . . . . . 10
Parameter NU . . . . . 0.2
Kernel Function. . . . . Linear
Norm of Longest Vector . . . . . 49.1966
Mean Squared Error (MSE, Training Data). . . . . 6.79961e-317
Mean Absolute Error (MAE, Training Data) . . . . 6.79961e-317
Average Loss (Training Data) . . . . . 6.79961e-317
Average Positive Loss (Training Data). . . . 0.0592752 (N=13)
Average Negative Loss (Training Data). . . . 0.0550412 (N=14)
Number Fast Runs through TrainData . . . . . 71
Number Slow Runs through TrainData . . . . . 73
Total Number of Kernel Calls . . . . . 1351
Time for Optimization. . . . . 426
Total Processing Time. . . . . 426
```

For some pairs (C, ν) the continuation algorithm was not successful (resulting in "too small step size in Newton"). But the best solution is very close to that found with PCx:

```
Grid Search Results: L1 Error (Training)
```

```
**************************************

         Dense Matrix (11 by 3)

     |       Nu0.2      Nu0.5      Nu0.8
   ------------------------------------
  c1 |   0.9629629  0.9629629  0.9629629
  c2 |   0.9629629  0.4589641  0.1558270
  c3 |   0.3659977  0.1211891     .
  c4 |   0.2266486  0.0666186     .
  c5 |   0.1593466     .          .
  c6 |   0.1147951     .          .
  c7 |      .          .          .
  c8 |   6.65e+010  8.37e+010     .
  c9 |      .       4.63e+010  2.73e+010
 c10 |   0.0035667     .          .
 c12 |      .          .          .


Grid Search Results: L2 Error (Training)
**************************************

         Dense Matrix (11 by 3)

     |       Nu0.2      Nu0.5      Nu0.8
   ------------------------------------
  c1 |   26.000000  26.000000  26.000000
  c2 |   26.000000  12.392031  4.2073293
  c3 |   9.8819377  3.2721056     .
  c4 |   6.1195125  1.7987017     .
  c5 |   4.3023582     .          .
  c6 |   3.0994686     .          .
  c7 |      .          .          .
  c8 |   1.80e+012  2.26e+012     .
  c9 |      .       1.25e+012  7.38e+011
 c10 |   0.0963013     .          .
 c12 |      .          .          .


FSM Grid Search: N Nonzero Coefficients
**************************************

         Dense Matrix (11 by 3)

           |   Nu0.2  Nu0.5  Nu0.8
         -------------------------
```

c1	0	0	0
c2	0	1	5
c3	1	6	0
c4	3	16	0
c5	7	0	0
c6	10	0	0
c7	0	0	0
c8	713	713	0
c9	0	713	713
c10	26	0	0
c12	0	0	0

```
Duality Gap. . . . . .
Geometric Margin . . . . .  6.79961e-317 (|w|^2= 6.79961e-317)
Number Support Vectors . . . . . .  0 (  0.00 %)
Number Support Vectors on Margin . . . . . 0
Bias . . . . . .
Radius of Sphere Around SV . . . . . .
Estimated VCdim of Classifier. . . . . .
Linear Kernel Constant (Fit) . . . . . -0.00614248
Linear Kernel Constant (PCE) . . . . . .
```

Nonzeros of Separating Plane (w*x = -0.00614248)

1	75	X75	0.149587812
2	253	X253	-0.036052715
3	281	X281	-0.018673456
4	332	X332	0.032446853
5	367	X367	-0.014687646
6	373	X373	-0.018673456
7	488	X488	0.068724321
8	489	X489	0.021404810
9	531	X531	-0.018673456
10	557	X557	-0.010719208
11	620	X620	0.000946335
12	630	X630	0.018321902
13	632	X632	-0.019322962
14	633	X633	-0.019322962
15	647	X647	0.009302708
16	649	X649	-0.008386675
17	650	X650	-0.008386675
18	652	X652	0.025737744
19	662	X662	0.061495011

```
20 679  X679  0.612910989
21 682  X682 -0.058670652
22 684  X684  0.112350138
23 686  X686  0.017331533
24 687  X687  0.048005754
25 689  X689 -0.131666456
26 699  X699  0.019490764


Largest 26 Plane Coefficients (Sorted)
**************************************

 1 679  X679  0.612910989
 2  75  X75   0.149587812
 3 689  X689 -0.131666456
 4 684  X684  0.112350138
 5 488  X488  0.068724321
 6 662  X662  0.061495011
 7 682  X682 -0.058670652
 8 687  X687  0.048005754
 9 253  X253 -0.036052715
10 332  X332  0.032446853
11 652  X652  0.025737744
12 489  X489  0.021404810
13 699  X699  0.019490764
14 632  X632 -0.019322962
15 633  X633 -0.019322962
16 281  X281 -0.018673456
17 373  X373 -0.018673456
18 531  X531 -0.018673456
19 630  X630  0.018321902
20 686  X686  0.017331533
21 367  X367 -0.014687646
22 557  X557 -0.010719208
23 647  X647  0.009302708
24 649  X649 -0.008386675
25 650  X650 -0.008386675
26 620  X620  0.000946335
Sparsity: 26 Nonzeros 687 Zeros (C=10)
```

Due to time consuming nonconverging runs for some pairs (C, ν) the computing time of the continuation algorithm is almost the same as with PCx. (Under normal conditions, the continuation algorithm is more than twice as fast as PCx for this dense LP problem.)

```
Total Number of Kernel Calls: 1351
    Time for Optimization: 426
    Total Processing Time: 427
```

Using the FQP method the common ν regression model is fitted and no sparse solution is found. Since the CACO2 data set contains more variables than observations, we always expect an exact fit for the entire grid table.

```
/* NUREG: meth=FQP */
c = [ 1. 2. 3. 4. 5. 6. 7. 10. 12. ];
nu = [ .2 .5 .8 ];
tun_g = c |> nu ;
modl = "714 = 1 : 713";
optn = [ "print"           2 ,
         "pplan"             ,
         "task"      "nureg" ,
         "meth"        "fqp" ,
         "kern"       "line" ];
alfa = svm(caco,modl,optn,.,tun_g);
```

```
     ****************************************************
     **********- Evaluation for Best C=1 Nu=0.2-**********
     ****************************************************

 Regularization Parameter C . . . . . . . . . . . . . . . . . . 1
 Parameter NU . . . . . . . . . . . . . . . . . . . . . . . . . 0.2
 Kernel Function. . . . . . . . . . . . . . . . . . .        Linear
 Norm of Longest Vector . . . . . . . . . . . . . . . . . 49.1864
 Mean Squared Error (MSE, Training Data). . . . .  1.24575e-027
 Mean Absolute Error (MAE, Training Data) . . . .  2.58913e-014
 Average Loss (Training Data) . . . . . . . . . .  2.58913e-014
 Average Positive Loss (Training Data). . . 2.68898e-014 (N=13)
 Average Negative Loss (Training Data). . . 2.49642e-014 (N=14)
 Number Fast Runs through TrainData . . . . . . . . . . . . 138
 Number Slow Runs through TrainData . . . . . . . . . . . . 114
 Total Number of Kernel Calls . . . . . . . . . . . . . . 12151
 Time for Optimization. . . . . . . . . . . . . . . . . . . . 2
 Total Processing Time. . . . . . . . . . . . . . . . . . . . 2
```

As expected we obtain perfect fit for all grid points:

Grid Search Results: L1 Error (Training)
****

Dense Matrix (9 by 3)

	Nu0.2	Nu0.5	Nu0.8
c1	1.25e-027	3.28e-027	1.01e-026
c2	4.37e-027	2.95e-026	4.89e-026
c3	1.18e-026	4.30e-026	9.87e-026
c4	1.12e-026	1.19e-025	1.69e-025
c5	2.87e-026	1.14e-025	5.06e-025
c6	3.55e-026	2.10e-025	4.45e-025
c7	3.62e-026	4.72e-025	9.59e-025
c10	1.21e-025	5.48e-025	1.84e-024
c12	1.32e-025	1.02e-024	1.68e-024

Grid Search Results: L2 Error (Training)
****

Dense Matrix (9 by 3)

	Nu0.2	Nu0.5	Nu0.8
c1	3.36e-026	8.85e-026	2.72e-025
c2	1.18e-025	7.96e-025	1.32e-024
c3	3.19e-025	1.16e-024	2.67e-024
c4	3.02e-025	3.22e-024	4.56e-024
c5	7.74e-025	3.09e-024	1.37e-023
c6	9.58e-025	5.66e-024	1.20e-023
c7	9.78e-025	1.28e-023	2.59e-023
c10	3.28e-024	1.48e-023	4.98e-023
c12	3.57e-024	2.76e-023	4.52e-023

```
Optimization Criterion . . . . . -0.0182032
Infinity Norm of Gradient. . . . . 4.15233e-012
Geometric Margin . . . . . 10.4819 (|w|^2= 0.0364064)
Number Support Vectors . . . . . 27 (100.00 %)
Number Support Vectors on Margin . . . . . 0
Bias . . . . . -2.56996e-018
Radius of Sphere Around SV . . . . . 49.1864
Estimated VCdim of Classifier. . . . . 89.0781
Linear Kernel Constant (Fit) . . . . . 0.0724425
Linear Kernel Constant (PCE) . . . . . .
```

Largest 100 Plane Coefficients (Sorted)

****

1	687	X687	0.032816801
2	686	X686	0.030054164
3	647	X647	0.026762444
4	649	X649	-0.026762437
5	650	X650	-0.026762437
6	679	X679	0.025326935
7	684	X684	0.024573302
8	682	X682	-0.023039144
9	489	X489	0.022891708
10	620	X620	0.022403264
11	699	X699	0.022063507
12	630	X630	0.021548100
13	632	X632	-0.021548098
14	633	X633	-0.021548098
15	708	X708	-0.021223372
16	709	X709	0.021223372
17	382	X382	-0.020925400
18	636	X636	0.019905032
19	411	X411	-0.019618004
20	662	X662	0.019566626
21	622	X622	0.019407145
22	75	X75	0.018557701
23	689	X689	-0.018312728
24	332	X332	0.017922034
25	253	X253	-0.016869153
26	627	X627	0.016866085
27	231	X231	0.016861994
28	696	X696	0.015689498
29	599	X599	0.015130573
30	674	X674	-0.014877561
31	660	X660	0.014704275
32	258	X258	-0.014668918
33	574	X574	-0.014488308
34	72	X72	-0.014469304
35	703	X703	0.014328720
36	712	X712	-0.014164813
37	615	X615	-0.014124916
38	515	X515	0.014096815
39	327	X327	0.014069182
40	270	X270	-0.014030942
41	121	X121	-0.013896892
42	499	X499	0.013829119
43	1	X1	0.013799132
44	32	X32	0.013609091
45	21	X21	-0.013293365
46	681	X681	-0.013221595
47	488	X488	0.013028779
48	624	X624	0.012868789
49	666	X666	0.012810016
50	704	X704	-0.012776149
51	555	X555	-0.012775061
52	176	X176	0.012761288
53	526	X526	-0.012570876
54	504	X504	0.012440034
55	183	X183	-0.012341057
56	367	X367	-0.012236605
57	188	X188	0.012213483
58	641	X641	-0.012174991
59	643	X643	-0.012174991
60	500	X500	0.012023641
61	149	X149	0.012012378
62	610	X610	-0.012009080
63	338	X338	0.011981737
64	264	X264	-0.011848599
65	517	X517	-0.011635896
66	525	X525	0.011571440
67	266	X266	-0.011428123
68	577	X577	0.011253424
69	655	X655	0.011183604
70	656	X656	0.011183604
71	30	X30	-0.011181298
72	281	X281	-0.011139274
73	373	X373	-0.011139274
74	531	X531	-0.011139274
75	276	X276	-0.011094159
76	690	X690	-0.010981108
77	150	X150	0.010917070
78	167	X167	-0.010841449
79	162	X162	0.010647111
80	78	X78	0.010643055

```
 81 680  X680 -0.010549018          91 626  X626 -0.010141162
 82 369  X369  0.010543959          92 598  X598  0.010106191
 83 607  X607 -0.010513646          93  60  X60  -0.010086264
 84 339  X339  0.010510089          94 147  X147  0.010052142
 85 511  X511 -0.010373241          95 677  X677  0.010041507
 86  62  X62   0.010250134          96 478  X478 -0.009958544
 87 671  X671 -0.010214544          97 401  X401  0.009950180
 88 562  X562  0.010208756          98 146  X146  0.009891022
 89 232  X232  0.010173118          99  68  X68  -0.009863944
 90 184  X184 -0.010148883         100 651  X651 -0.009697682
                                Sparsity: 713 Nonzeros 0 Zeros (C=1)


                Total Number of Kernel Calls: 12151
                     Time for Optimization: 2
                     Total Processing Time: 2
```

3. Example for stepwise feature selection with SV classification:

   A modified form of the algorithm by Guyon et al. (2002) is performed. NIR Spectra data set: train: nr=21, test: nr=7, nvar= 268. Use median of y for binary target.

```
  print "NIR Spectra data set: train: nr=21, test: nr=7";
  options NOECHO;
#include "..\\tdata\\nir.dat"
  options ECHO;
  nr = nrow(xtrn); nc = ncol(xtrn);
  print "nrtrn,nctrn=",nr,nc;


  sopt = [ "ari" "std" "med" ];
  mom = univar(ytrn',sopt);
  /* print "Moments of y=", mom; */
  cutof = mom[3];
  y = ytrn .< cutof; tnir = xtrn -> y'; /* attrib(tnir); */
  y = ytst .< cutof; test = xtst -> y'; /* attrib(test); */
  /* print tnir[ , 200:269 ]; */
  /* model: */
  xind = [ 1:268 ]; yind = 269;
  modl = "269 = 1 : 268";
  class = 269;


  /* Random CROSS validation: use fold=10: FQP */
```

```
optn = [ "print"           3 ,
         "pplan"             ,
         "task"       "sfsc" ,
         "meth"        "fqp" ,
         "nvarmin"         3 ,
         "vardrop"        .2 ,
         "fold"           10 ,
         "c"              1. ,
         "kern"       "line" ];
alfa = svm(tnir,modl,optn,class);
```

```
                    *****************
                    Model Information
                    *****************

                Number Valid Observations   21
                Response Variable        Y[269]
                N Independend Variables    268
                SV PatternSearch
                Model with 1 Linear Constraint
                Estimation Method          FQP
                Optimization Method     QPNUSP
                Kernel Function         Linear
                Block CV                    10
                CV Evaluated at Test Data
                Use Unscaled Predictor


                   ***********************
                   Class Level Information
                   ***********************

 Class  Level    Value

Y[  1]      2        0       1


            ************************************
            Number of Observations for Class Levels
            ************************************

            Variable       Value    Nobs  Proportion
            Y[269]             0      11   52.380952
                               1      10   47.619048
```

```
*******************************************
Stepwise Variable Selection (Ntrn=21 Nvar=268)
   Size: Trainig Data=10 Validation Data=11
*******************************************
```

Iter	NDrop	WgtDrop	NRem	WgtRemain	MiscTrn
1	0	.	268	11.546638	.
2	54	0.3202434	214	11.334699	.
3	43	0.6239116	171	7.5385469	.
4	34	0.3093094	137	10.523981	.
5	27	1.2406915	110	8.0957709	.
6	22	0.7713951	88	5.4381890	.
7	18	0.5859408	70	5.9357352	.
8	14	0.6946530	56	4.1798109	.
9	11	0.5979633	45	3.6850981	.
10	9	0.5665638	36	3.5387044	.
11	7	0.5319841	29	2.4866877	.
12	6	0.3723136	23	2.3945302	.
13	5	0.4540156	18	2.0681687	.
14	4	0.3959728	14	1.7774829	.
15	3	0.3275218	11	0.5769029	.
16	1	0.0398765	10	1.3806106	.
17	2	0.2123163	8	2.2412667	1
18	2	0.3114333	6	3.9698555	2
19	1	0.4859327	5	2.2713750	1
20	1	0.2374750	4	3.7636927	1
21	1	0.6223037	3	1.6570000	1

```
*******************************
Evaluation of Training Data Fit
*******************************
```

Index	Value	StdErr
Absolute Classification Error	4	.
Classification Accuracy	80.95238095	.
Concordant Pairs	65.45454545	.
Discordant Pairs	3.636363636	.
Tied Pairs	30.90909091	.
Goodman-Kruskal Gamma	0.894736842	0.110872555
Kendall Tau_b	0.618181818	0.171732594
Stuart Tau_c	0.616780045	0.171914534
Somers D C|R	0.618181818	0.171824522

Classification Table

Observed	Predicted 0	1
0	9	2
1	2	8

```
Regularization Parameter C . . . . . 1
Kernel Function. . . . . Linear
Norm of Longest Vector . . . . . 31.955
Number Misclassifactions (Training Data) . . . . . 4
Number Misclassifactions (Block Cross Val.). . . . . 0
Number Fast Runs through TrainData . . . . . 476
Number Slow Runs through TrainData . . . . . 478
Total Number of Kernel Calls . . . . . 3287
Time for Optimization. . . . . 0
Total Processing Time. . . . . 0
Optimization Criterion . . . . . -13.5267
Infinity Norm of Gradient. . . . . 1.77636e-015
Geometric Margin . . . . . 6.79961e-317 (|w|^2= 6.79961e-317)
Number Support Vectors . . . . . 16 ( 76.19 %)
Number Support Vectors on Margin . . . . . 14
Bias . . . . . 1.38778e-017
Radius of Sphere Around SV . . . . . 4.38068
Estimated VCdim of Classifier. . . . . .
Linear Kernel Constant (Fit) . . . . . 6.00227
Linear Kernel Constant (PCE) . . . . . 5.74027

Number Block CV Computations . . . . . 0
CV Estimate of Error (Nmis=0). . . . . 6.79961e-317 %
CV Estimate of Recall. . . . . 6.79961e-317 %
CV Estimate of Precision . . . . . 6.79961e-317 %
```

Nonzeros of Separating Plane (w*x = 6.00227)

1	53	X53	0.049347450
2	71	X71	-2.071485192
3	129	X129	-0.750701791

Largest 3 Plane Coefficients (Sorted)

```
************************************

        1  71  X71  -2.071485192
        2 129  X129 -0.750701791
        3  53  X53   0.049347450
Sparsity: 3 Nonzeros 265 Zeros (C=1)


 Total Number of Kernel Calls: 3287
      Time for Optimization: 0
      Total Processing Time: 0


/* Random CROSS validation: use fold=10: ASVM */
optn = [ "print"           3 ,
         "pplan"             ,
         "task"       "sfsc" ,
         "meth"       "asvm" ,
         "nvarmin"         3 ,
         "vardrop"        .2 ,
         "fold"           10 ,
         "c"              1. ,
         "kern"       "line" ];
alfa = svm(tnir,modl,optn,class);


         ******************************************
         Stepwise Variable Selection (Ntrn=21 Nvar=268)
            Size: Trainig Data=10 Validation Data=11
         ******************************************

        Iter  NDrop    WgtDrop   NRem  WgtRemain  MiscTrn
          1      0     .          268  11.546638        .
          2     54  0.3202434    214  11.334699        .
          3     43  0.6239116    171  7.5385469        .
          4     34  0.3093094    137  10.523981        .
          5     27  1.2406915    110  8.0957709        .
          6     22  0.7713951     88  5.4381890        .
          7     18  0.5859408     70  5.9357352        .
          8     14  0.6946530     56  4.1798109        .
          9     11  0.5979633     45  3.6850981        .
         10      9  0.5665638     36  3.5387044        .
         11      7  0.5319841     29  2.4866877        .
         12      6  0.3723136     23  2.3945302        .
         13      5  0.4540156     18  2.0681687        .
         14      4  0.3959728     14  1.7774829        .
```

15	3	0.3275218	11	0.5769029	.
16	1	0.0398765	10	1.3806106	.
17	2	0.2123163	8	3.0443795	2
18	2	0.3303509	6	2.9416886	2
19	1	0.3412764	5	2.7598094	1
20	1	0.4252886	4	2.2291107	1
21	1	0.4825829	3	1.9271501	2

ASVM Iteration History

iter	nact	crit	gmax	absxdif	relxdif
1	0	-0.05050774	1.00000000	0.00000000	0.00000000

****
Evaluation of Training Data Fit
****

Index	Value	StdErr
Absolute Classification Error	3	.
Classification Accuracy	85.71428571	.
Concordant Pairs	73.63636364	.
Discordant Pairs	1.818181818	.
Tied Pairs	24.54545455	.
Goodman-Kruskal Gamma	0.951807229	0.061721066
Kendall Tau_b	0.718181818	0.149438434
Stuart Tau_c	0.716553288	0.150478923
Somers D C|R	0.718181818	0.150078867

Classification Table

	Predicted	
Observed	0	1
0	9	2
1	1	9

Regularization Parameter C 1
Kernel Function. . . . . Linear
Norm of Longest Vector 31.9706
Number Misclassifactions (Training Data) 3

```
Number Misclassifactions (Block Cross Val.). . . . . 0
Number Fast Runs through TrainData . . . . . 786
Number Slow Runs through TrainData . . . . . 839
Total Number of Kernel Calls . . . . . 316
Time for Optimization. . . . . 1
Total Processing Time. . . . . 1
Optimization Criterion . . . . . -0.0505077
Infinity Norm of Gradient. . . . . 1
Geometric Margin . . . . . 6.79961e-317 (|w|^2= 6.79961e-317)
Number Support Vectors . . . . . 21 (100.00 %)
Number Support Vectors on Margin . . . . . 4
Bias . . . . . 1.00481
Radius of Sphere Around SV . . . . . 4.46786
Estimated VCdim of Classifier. . . . . .
Linear Kernel Constant (Fit) . . . . . 0.101015
Linear Kernel Constant (PCE) . . . . . .

Number Block CV Computations . . . . . 0
CV Estimate of Error (Nmis=0). . . . . 6.79961e-317 %
CV Estimate of Recall. . . . . 6.79961e-317 %
CV Estimate of Precision . . . . . 6.79961e-317 %


        Nonzeros of Separating Plane (w*x = 0.101015)
        ---------------------------------------------

                1  52  X52   0.680303476
                2  71  X71  -1.285130677
                3 129  X129 -0.057619822


            Largest 3 Plane Coefficients (Sorted)
            *************************************

                1  71  X71  -1.285130677
                2  52  X52   0.680303476
                3 129  X129 -0.057619822
            Sparsity: 3 Nonzeros 265 Zeros (C=1)


              Total Number of Kernel Calls: 316
                  Time for Optimization: 1
                  Total Processing Time: 1


/* Random CROSS validation: use fold=10: PSVM */
```

```
optn = [ "print"           3 ,
         "pplan"             ,
         "task"       "sfsc" ,
         "meth"       "psvm" ,
         "nvarmin"         3 ,
         "vardrop"        .2 ,
         "fold"           10 ,
         "c"              1. ,
         "kern"       "line" ];
alfa = svm(tnir,modl,optn,class);
```

```
         ******************************************
         Stepwise Variable Selection (Ntrn=21 Nvar=268)
            Size: Trainig Data=10 Validation Data=11
         ******************************************
```

Iter	NDrop	WgtDrop	NRem	WgtRemain	MiscTrn
1	0	.	268	11.546638	.
2	54	0.3202434	214	11.334699	.
3	43	0.6239116	171	7.5385469	.
4	34	0.3093094	137	10.523981	.
5	27	1.2406915	110	8.0957709	.
6	22	0.7713951	88	5.4381890	.
7	18	0.5859408	70	5.9357352	.
8	14	0.6946530	56	4.1798109	.
9	11	0.5979633	45	3.6850981	.
10	9	0.5665638	36	3.5387044	.
11	7	0.5319841	29	2.4866877	.
12	6	0.3723136	23	2.3945302	.
13	5	0.4540156	18	2.0681687	.
14	4	0.3959728	14	1.7774829	.
15	3	0.3275218	11	0.5769029	.
16	1	0.0398765	10	1.3806106	.
17	2	0.2123163	8	3.0443795	2
18	2	0.3303509	6	2.9416886	2
19	1	0.3412764	5	2.7598094	1
20	1	0.4252886	4	2.2291107	1
21	1	0.4825829	3	1.9271501	2

```
              ******************************
              Evaluation of Training Data Fit
              ******************************
```

Index	Value	StdErr

Absolute Classification Error	3	.
Classification Accuracy	85.71428571	.
Concordant Pairs	73.63636364	.
Discordant Pairs	1.818181818	.
Tied Pairs	24.54545455	.
Goodman-Kruskal Gamma	0.951807229	0.061721066
Kendall Tau_b	0.718181818	0.149438434
Stuart Tau_c	0.716553288	0.150478923
Somers D C|R	0.718181818	0.150078867

Classification Table

Observed |	Predicted 0	1
0	9	2
1	1	9

Regularization Parameter C	1
Kernel Function.	Linear
Norm of Longest Vector	31.9706
Number Misclassifactions (Training Data)	3
Number Misclassifactions (Block Cross Val.)	0
Number Fast Runs through TrainData	578
Number Slow Runs through TrainData	631
Total Number of Kernel Calls	316
Time for Optimization.	0
Total Processing Time.	0
Optimization Criterion	-0.0505077
Infinity Norm of Gradient.	1
Geometric Margin	6.79961e-317 (|w|^2= 6.79961e-317)
Number Support Vectors	21 (100.00 %)
Number Support Vectors on Margin	4
Bias	1.00481
Radius of Sphere Around SV	4.46786
Estimated VCdim of Classifier.	.
Linear Kernel Constant (Fit)	0.101015
Linear Kernel Constant (PCE)	.
Number Block CV Computations	0
CV Estimate of Error (Nmis=0).	6.79961e-317 %
CV Estimate of Recall.	6.79961e-317 %
CV Estimate of Precision	6.79961e-317 %

```
Nonzeros of Separating Plane (w*x = 0.101015)
---------------------------------------------

           1  52  X52   0.680303476
           2  71  X71  -1.285130677
           3 129  X129 -0.057619822


    Largest 3 Plane Coefficients (Sorted)
    *************************************

           1  71  X71  -1.285130677
           2  52  X52   0.680303476
           3 129  X129 -0.057619822
    Sparsity: 3 Nonzeros 265 Zeros (C=1)


      Total Number of Kernel Calls: 316
          Time for Optimization: 0
          Total Processing Time: 0
```

The illustration section at the end of this newsletter shows some applications of the `svm` function on a very large set of micro array data with multinomial response.

2.5 New Functions

New functions are implemented:

garotte implements the *Garotte* algorithm of constrained linear regression by Breiman (1993).

generead not finished yet.

lars implements *least angle regression*, including the *lasso* and *forward stagewise*

nlkpca nonlinear Hilbert kernel principal components

nlkpls nonlinear Hilbert kernel partial least squares

3 New Developments

3.1 Function `garotte`

< beta,shrnk,yptrn,yptst > = garotte(trn,model,optn<,class<,test>>)

Purpose: The *garotte* is a simple method for variable (feature) selection by constrained least squares (Breiman, 1993). It estimates the n weights $c_j \geq 0$ of the L_1 length restricted linear least squares regression problem

$$\min_c \sum_i^N (y_i - \sum_j^n x_{ij} c_j \hat{\beta}_j^0)^2,$$

$$\text{s.t.} \quad \sum_j^n c_j \leq t, \quad \text{and} \quad c_j \geq 0, \quad j = 1, \ldots, n$$

where $\hat{\beta}_j^0$, $\hat{\alpha}^0$ are the estimates of the unconstrained linear least squares regression:

$$\min_\beta (y - \mathbf{X}\beta)^T (y - \mathbf{X}\beta)$$

and where $t < t_0 = \sum_j^n |\hat{\beta}_j^0|$ and $\mathbf{X}$ and y are the mean centered data. Note, the method works only when the unconstrained LS estimates are available, i.e. for nonsingular design matrix $\mathbf{X}^T\mathbf{X}$. It requires that the number of variables is not larger than the number of observations. The problem is similar to that of the Lasso (Tibshirani,1996) and LARS and that of ridge regression (which is using an L_2 rather than L_1 penalty term).

Input: **trn** the $N \times M$ input data set where the M columns contain the $N \times n$ predictor matrix $\mathbf{X}$ and the N response vector $\mathbf{Y}$. Which data columns are selected for $\mathbf{X}$ and $\mathbf{Y}$ must be specified with the model string. Before computing the *garotte* estimates, the data $(\mathbf{X}, Y)$ will be mean centered.

model : The analysis model is specified in form of a string, e.g. `model= "3=1 2"`, containing column numbers for variables. The syntax of the `model` string argument is the same as for the `glmod()` function except for the additional *events / trial* response specification. **????**

optn : The option argument is specified in form of a simple vector.

Index	Meaning
[1]	ipri: amount of printed output
[2]	t: the nonnegative garotte coefficients add to this absolute value ($0 < t \leq t_0$)
[3]	prop: $t = prop * t_0$ the nonnegative garotte coefficients add to this portion of the absolute sum t_0 of LS coefficients ($0 < prop \leq 1$, default $prop = .75$)
[4]	atol: precision of estimates (default=.0001)

Only one of the options `[2]` or `[3]` should be specified and the other should be missing. The absolute value `[2]` is more difficult and would require to solve the unconstrained LS problem first. The relative value `[3]` seems to be easier and should be a factor smaller than one. Note, that the *garotte* model is defined for centered data and that there is no intercept estimate, which makes the `noint` option irrelevant.

class : This optional argument should be an integer scalar or vector of integer scalars naming the number of columns which are considered categorical (nominal scaled) variables.

test specifies a matrix of test data which is not used for modeling but for predicted values are computed (scored) that can be returned with the last output argument `yptt`.

Output: **beta** vector with least squares parameter estimates $\hat{\beta}_j^0$ of centered data (without intercept estimate)

shrnk the vector of nonnegative c_j shrink coefficients

yptrn $2 \times N$ matrix with predicted values and residuals for training data

yptst $2 \times N(test)$ matrix with predicted values and residuals for $N(test)$ test data

Note, the parameter estimates of the *garotte* model are `parm= beta .* shrk;` .

Restrictions: 1. The input data set should not contain missing values, string or complex data.

2. The model was improved by the *Lasso* (Tibshirani, 1996; Efron et al., 2002). See the `lars()` function.

Relationships: lars(), lls(), reg(), pls()

Examples: 1. Australian Institute of Sport (ais) data set (Weisberg, 2002): There are $N = 202$ observations and usually only four of the variables are used for predictors.

```
options NOECHO;
ais = [
%inc "..\\tdata\\ais.dat";
];
options ECHO;
cname = [ "Sex" "Ht" "Wt" "LBM" "RCC" "WCC" "Hc" "Hg" ];
ais = shape(ais,.,8);
nobs = nrow(ais); nvar = ncol(ais);
ais = ais -> log(ais[,5]);
```

```
optn = [ 3 ,   /* ipri */
         . ,   /* sumb */
       .05 ];  /* prop */
modl = "4 = 2 3 9 6";
< lsest,garest,yptr > = garotte(ais,modl,optn);
```

```
                  *****************
                  Model Information
                  *****************

               Number Valid Observations  202
               Response Variable          Y[4]
               N Independend Variables      4
               Number Parameters            5
               Proportion of NN Coe 0.0500000
               Linear Model Without Intercept


                    *************
                    Model Effects
                    *************

                  X2 + X3 + X9 + X6


                  *****************
                  Simple Statistics
                  *****************

 Column  Nobs        Mean     Std Dev    Skewness    Kurtosis
 Y[4]     202   64.873713   13.070197   0.3611965  -0.2158696
 X[2]     202   180.10396   9.7344945  -0.2007970   0.5717271
 X[3]     202   75.008168   13.925574   0.2424318   0.4254397
 X[9]     202   1.5468840   0.0963023   0.1185203  -0.1197264
 X[6]     202   7.1086634   1.8005490   0.8413109   1.5149690

Least-Sqares Solution (No Intercept, Centered Data)
---------------------------------------------------

  1 :      0.2159     0.6834      28.43    -0.4006

Nonnegative Garotte Coefficients
--------------------------------
```

```
 1 :           0      1.187     0.2993          0


                  Number Nonzeros in Garotte= 2
              Precision of Garotte Solution= 0.000136201
                  Number Sweeps of 4 by 4 Matrix= 11
                       Total Processing Time: 0
```

2. Boston Housing Data: nobs=506, nvar=14:

```
fid = fopen("..\\tdata\\housing.dat","r");
form = "%g %g %g %g %g %g %g %g %g %g %g %g %g %g";
hous = fscanf(fid,form,506,14);
vnam = [ "crim" "zn" "indus" "chas" "nox" "rm" "age"
         "dis" "rad" "tax" "ptrat" "b" "lstat" "medv" ];
hous = cname(hous,vnam);


 optn = [ 3 ,   /* ipri */
          . ,   /* sumb */
         .1 ];  /* prop */
 modl = "14 = 1:13";
 < lsest,garest,yptr > = garotte(hous,modl,optn);


                        *****************
                        Model Information
                        *****************

                  Number Valid Observations  506
                  Response Variable         Y[14]
                  N Independend Variables     13
                  Number Parameters           14
                  Proportion of NN Coe 0.1000000
                  Linear Model Without Intercept


                          *************
                          Model Effects
                          *************

X1  + X2  + X3  + X4  + X5  + X6  + X7  + X8  + X9  + X10 + X11 +
                            X12 + X13


                        *****************
                        Simple Statistics
```

```
*****************

 Column   Nobs       Mean     Std Dev    Skewness    Kurtosis
 Y[14]     506   22.532806   9.1971041   1.1080984   1.4951969
 X[ 1]     506   3.6135236   8.6015451   5.2231487   37.130508
 X[ 2]     506   11.363636   23.322453   2.2256663   4.0315101
 X[ 3]     506   11.136779   6.8603530   0.2950215  -1.2335397
 X[ 4]     506   0.0691700   0.2539940   3.4059042   9.6382638
 X[ 5]     506   0.5546951   0.1158777   0.7293079  -0.0646673
 X[ 6]     506   6.2846344   0.7026172   0.4036121   1.8915002
 X[ 7]     506   68.574901   28.148862  -0.5989626  -0.9677156
 X[ 8]     506   3.7950427   2.1057101   1.0117806   0.4879412
 X[ 9]     506   9.5494071   8.7072594   1.0048146  -0.8672320
 X[10]     506   408.23715   168.53712   0.6699559  -1.1424080
 X[11]     506   18.455534   2.1649458  -0.8023248  -0.2850920
 X[12]     506   356.67403   91.294863  -2.8903738   7.2268177
 X[13]     506   12.653063   7.1410615   0.9064601   0.4932395

Least-Sqares Solution (No Intercept, Centered Data)
---------------------------------------------------

  1 :      -0.108    0.04642    0.02056      2.687     -17.77
  6 :        3.81  0.0006922     -1.476      0.306   -0.01233
 11 :     -0.9527   0.009312    -0.5248

Nonnegative Garotte Coefficients
--------------------------------

  1 :           0          0          0          0          0
  6 :       1.011          0    0.01992          0          0
 11 :      0.6219          0      1.119

            Number Nonzeros in Garotte= 4
          Precision of Garotte Solution= 0.000226266
             Number Sweeps of 13 by 13 Matrix= 22
                   Total Processing Time: 0
```

3.2 Function generead

3.3 Function lars

< gof,beta,yptrn,yptst > = lars(trn,model,optn<,class<,test>>)

Purpose: The `lars()` function incorporates a number of special cases of the L_1 and L_2 penalized linear regression problem

$$\min_{\beta} \sum_{i}^{N} (y_i - \sum_{j}^{n} x_{ij}\hat{\beta}_j)^2 + \lambda_1 \sum_{j}^{n} |\beta_j| + \lambda_2 \sum_{j}^{n} ||\beta_j||^2$$

which is called the *elastic net* by Zou & Hastie (2003). The solution $\hat{\beta}$ of the *elastic net* depends on the two hyperparameters (λ_1, λ_2). The following methods are implemented in the `lars()` function:

1. Least Angle Regression (LARS): $\lambda_2 = 0$ (Efron et al., 2002)
2. Lasso: $\lambda_2 = 0$ (Tibshirani, 1996)
3. Foreward Stagewise: $\lambda_2 = 0$ (Efron et al., 2002)
4. Ridge Regression $\lambda_1 = 0$
5. Elastic Net $(\lambda_1 \geq 0, \lambda_2 \geq 0)$
6. Univariate Soft Thresholding (ULS) $(\lambda > 0, \lambda_2 \to \infty)$ (Donoho et al., 1995)

The Lasso introduced by Tibshirani (1993) is a modification of the earlier *Garotte* by Breiman (1993) and estimates the n parameters β_j of the L_1 length restricted linear least squares regression problem

$$\min_{\beta} \sum_{i}^{N} (y_i - \sum_{j}^{n} x_{ij}\hat{\beta}_j)^2 + \lambda_1 \sum_{j}^{n} |\beta_j|$$

which is equivalent to the problem

$$\min_{\beta} \sum_{i}^{N} (y_i - \sum_{j}^{n} x_{ij}\hat{\beta}_j)^2, \quad \text{s.t.} \quad \sum_{j}^{n} |\beta_j| \leq t$$

where $t < t_0 = \sum_j^n |\hat{\beta}_j^0|$ and where $\hat{\beta}_j^0$ are the estimates of the unconstrained linear least squares regression:

$$\min_{\beta} (y - \mathbf{X}\beta)^T (y - \mathbf{X}\beta)$$

and where $\mathbf{X}$ and y are the mean centered data. Enforcing the L_1 length constraint ensures that the optimal parameter vector $\hat{\beta}$ will have different degrees of sparsity depending on the treshold t. Very small values of t will result in zero vectors $\hat{\beta}$. Values of $t \geq t_0$ will usually result in dense vectors $\hat{\beta}$.

Input: **trn** the $N \times M$ input data set containing the $N \times m$ predictor matrix $\mathbf{X}$ and the N response vector $\mathbf{Y}$. Which data columns are selected for $\mathbf{X}$ and $\mathbf{Y}$ must be specified with the model string. Before computing the *garotte* estimates, the data $(\mathbf{X}, Y)$ will be mean centered.

model : The analysis model is specified in form of a string, e.g. `model= "3=1 2"`, containing column numbers for variables. The syntax of the `model` string argument is the same as for the `glmod()` function except for the additional *events / trial* response specification. `????`

optn : The option argument is specified in form of a simple vector.

Index	Meaning
[**1**]	ipri: amount of printed output
[**2**]	λ_2 is L_2 penalty for ridge regression
[**3**]	λ_1 is L_1 penmalty for Lasso
[**4**]	type=1: LARS: least angle regression;
	type=2: LASSO (default);
	type=3: forward stagewise;
	type=4: ridge regression (lam2);
	type=5: elastic net (lam1 and lam2);
	type=6: univariate soft thresholding (UST)
[**5**]	igrm=1: use $\mathbf{X}^T\mathbf{X}$ for Cholesky update (default for $n \leq N$)
	igrm=0: use QR decomposition for Cholesky update (default for $n > N$)
[**6**]	mxnzr : maximum number nonzero coefficients

class : This optional argument should be an integer scalar or vector of integer scalars naming the number of columns which are considered categorical (nominal scaled) variables.

test specifies a matrix of test data which is not used for modeling but for predicted values are computed (scored) that can be returned with the last output argument `yptt`.

Output: gof matrix of goodness-of-fit measures

beta matrix with (sparse) coefficients $\hat{\beta}$ of centered data (without intercept estimate)

yptrn $N_{step} \times N$ matrix with predicted values for training data

yptst $N_{step} \times N(test)$ matrix with predicted values for $N(test)$ test data

Restrictions: 1. The input data set should not contain missing values, string or complex data.

2.

Relationships: garotte(), lls(), reg(), pls()

Examples: 1. Diabetes Data set from LARS Package in R: nobs=442, nvar=10

```
print "Diabetes Data set from LARS Package in R";
options NOECHO;
DiabX = [
%inc "..\\tdata\\diabX.dat";
```

```
];
DiabX = shape(DiabX,.,10);
DiabY = [
%inc "..\\tdata\\diabY.dat";
];
options ECHO;
print "nrX,nrY=", nrow(DiabX),ncol(DiabY);


Diab = DiabX -> DiabY‘;
nr = nrow(Diab); nc = ncol(Diab);
print "nr,nc=", nr,nc;
cnam = [ "X1":"X10" "Y" ];
Diab = cname(Diab,cnam);
```

LASSO:

```
optn = [ 3 ];
modl = "11 = 1:10";
< gof,parm,yptr > = lars(Diab,modl,optn);
```

```
                    *****************
                    Model Information
                    *****************

                Number Valid Observations  442
                Response Variable         Y[11]
                N Independend Variables     10
                Number Parameters           11
                Linear Model Without Intercept


                      *************
                      Model Effects
                      *************

  X1  + X2  + X3  + X4  + X5  + X6  + X7  + X8  + X9  + X10


                    *****************
                    Simple Statistics
                    *****************

Column   Nobs        Mean     Std Dev    Skewness    Kurtosis
Y[11]     442   152.13348   77.093005   0.4405629  -0.8830573
X[ 1]     442  -2.94e-011   0.0476190  -0.2313815  -0.6712237
```

X[2]	442	-1.27e-009	0.0476190	0.1273845	-1.9928110
X[3]	442	2.74e-010	0.0476190	0.5981485	0.0950943
X[4]	442	3.53e-012	0.0476190	0.2906639	-0.5327797
X[5]	442	-5.43e-011	0.0476190	0.3781082	0.2329481
X[6]	442	8.29e-011	0.0476190	0.4365919	0.6013814
X[7]	442	-2.26e-011	0.0476190	0.7992551	0.9815077
X[8]	442	9.54e-010	0.0476190	0.7353736	0.4444011
X[9]	442	-1.60e-010	0.0476190	0.2917739	-0.1343659
X[10]	442	-8.24e-010	0.0476190	0.2079165	0.2369162

****
Iteration History: LASSO
****

Iter	Nact	Add	Dropp	gamma	arclen	R2
1	1	3	.	0.06332106	60.1192426	0.04217625
2	2	9	.	0.49073113	513.223759	0.35125417
3	3	4	.	0.30211221	175.553216	0.41733717
4	4	7	.	0.58828998	259.367461	0.47892806
5	5	2	.	0.31774527	88.6592445	0.49480593
6	6	10	.	0.22321055	43.6778396	0.50059988
7	7	5	.	0.71027043	135.984058	0.51341086
8	8	8	.	0.72586943	54.0156328	0.51536487
9	9	6	.	0.07089060	5.56733837	0.51568667
10	9	1	.	0.57119680	41.9995329	0.51737015
11	9	.	7	0.39950262	7.27071528	0.51745095
12	10	7	.	1.00000000	27.9700803	0.51774943

****
Iteration History: LASSO
****

Iter	Nact	Add	Dropp	SSE	R2	Cp
0	0	.	.	2621009.12	0.00000000	449.579049
1	1	3	.	2510464.79	0.04217625	414.059941
2	2	9	.	1700368.75	0.35125417	141.110702
3	3	4	.	1527164.60	0.41733717	84.3246495
4	4	7	.	1365734.31	0.47892806	31.5346821
5	5	2	.	1324118.28	0.49480593	19.4100637
6	6	10	.	1308932.27	0.50059988	16.2558840
7	7	5	.	1275354.57	0.51341086	6.85950364
8	8	8	.	1270233.11	0.51536487	7.12126109
9	9	6	.	1269389.65	0.51568667	8.83498873

10	9	1	.	1264977.24	0.51737015	9.33740339
11	9	.	7	1264765.46	0.51745095	11.2655238
12	10	7	.	1263983.13	0.51774943	13.0000000

Number Updates= 11
Number Downdates= 1
Total Processing Time: 0

Parameter Estimates (LASSO)
****

Dense Matrix (13 by 10)

	1	2	3	4	5
1	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2	0.0000000	0.0000000	60.119242	0.0000000	0.0000000
3	0.0000000	0.0000000	361.89461	0.0000000	0.0000000
4	0.0000000	0.0000000	434.75795	79.236444	0.0000000
5	0.0000000	0.0000000	505.65955	191.26988	0.0000000
6	0.0000000	-74.916586	511.34807	234.15466	0.0000000
7	0.0000000	-111.97854	512.04408	252.52702	0.0000000
8	0.0000000	-197.75647	522.26484	297.15973	-103.94623
9	0.0000000	-226.13365	526.88546	314.38927	-195.10587
10	0.0000000	-227.17580	526.39058	314.95048	-237.34182
11	-5.7189254	-234.39760	522.64878	320.34255	-554.26621
12	-7.0112248	-237.10077	521.07512	321.54903	-580.43853
13	-10.012180	-239.81908	519.83978	324.39043	-792.18454

	6	7	8	9	10
1	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
3	0.0000000	0.0000000	0.0000000	301.77537	0.0000000
4	0.0000000	0.0000000	0.0000000	374.91586	0.0000000
5	0.0000000	-114.10098	0.0000000	439.66496	0.0000000
6	0.0000000	-169.71144	0.0000000	450.66748	0.0000000
7	0.0000000	-196.04544	0.0000000	452.39276	12.078126
8	0.0000000	-223.92602	0.0000000	514.74950	54.767648
9	0.0000000	-152.47720	106.34288	529.91606	64.487389
10	33.628919	-134.59895	111.38430	545.48293	64.606642
11	286.73603	0.0000000	148.90052	663.03326	66.330916
12	313.86205	0.0000000	139.85792	674.93661	67.179361
13	476.74609	101.04479	177.06432	751.27948	67.625345

LARS:

```
optn = [ 3 , ., ., 1 ];
modl = "11 = 1:10";
< gof,parm,yptr > = lars(Diab,modl,optn);
```

****
Iteration History: LARS
****

Iter	Nact	Add	Dropp	gamma	arclen	R2
1	1	3	.	0.06332106	60.1192426	0.04217625
2	2	9	.	0.49073113	513.223759	0.35125417
3	3	4	.	0.30211221	175.553216	0.41733717
4	4	7	.	0.58828998	259.367461	0.47892806
5	5	2	.	0.31774527	88.6592445	0.49480593
6	6	10	.	0.22321055	43.6778396	0.50059988
7	7	5	.	0.71027043	135.984058	0.51341086
8	8	8	.	0.72586943	54.0156328	0.51536487
9	9	6	.	0.07089060	5.56733837	0.51568667
10	10	1	.	1.00000000	73.5290058	0.51774943

****
Iteration History: LARS
****

Iter	Nact	Add	Dropp	SSE	R2	Cp
0	0	.	.	2621009.12	0.00000000	453.726270
1	1	3	.	2510464.79	0.04217625	418.032248
2	2	9	.	1700368.75	0.35125417	143.801195
3	3	4	.	1527164.60	0.41733717	86.7410814
4	4	7	.	1365734.31	0.47892806	33.6956830
5	5	2	.	1324118.28	0.49480593	21.5052155
6	6	10	.	1308932.27	0.50059988	18.3270070
7	7	5	.	1275354.57	0.51341086	8.87749667
8	8	8	.	1270233.11	0.51536487	9.13115042
9	9	6	.	1269389.65	0.51568667	10.8435435
10	10	1	.	1263983.13	0.51774943	11.0000000

Number Updates= 10
Number Downdates= 0

```
                    Total Processing Time: 0

                  Parameter Estimates (LARS)
                  **************************

                    Dense Matrix (11 by 10)

   |           1          2          3          4          5
---+-----------------------------------------------------------
 1 |   0.0000000  0.0000000  0.0000000  0.0000000  0.0000000
 2 |   0.0000000  0.0000000  60.119242  0.0000000  0.0000000
 3 |   0.0000000  0.0000000  361.89461  0.0000000  0.0000000
 4 |   0.0000000  0.0000000  434.75795  79.236444  0.0000000
 5 |   0.0000000  0.0000000  505.65955  191.26988  0.0000000
 6 |   0.0000000 -74.916586  511.34807  234.15466  0.0000000
 7 |   0.0000000 -111.97854  512.04408  252.52702  0.0000000
 8 |   0.0000000 -197.75647  522.26484  297.15973 -103.94623
 9 |   0.0000000 -226.13365  526.88546  314.38927 -195.10587
10 |   0.0000000 -227.17580  526.39058  314.95048 -237.34182
11 |  -10.012180 -239.81908  519.83978  324.39043 -792.18454

   |           6          7          8          9         10
---+-----------------------------------------------------------
 1 |   0.0000000  0.0000000  0.0000000  0.0000000  0.0000000
 2 |   0.0000000  0.0000000  0.0000000  0.0000000  0.0000000
 3 |   0.0000000  0.0000000  0.0000000  301.77537  0.0000000
 4 |   0.0000000  0.0000000  0.0000000  374.91586  0.0000000
 5 |   0.0000000 -114.10098  0.0000000  439.66496  0.0000000
 6 |   0.0000000 -169.71144  0.0000000  450.66748  0.0000000
 7 |   0.0000000 -196.04544  0.0000000  452.39276  12.078126
 8 |   0.0000000 -223.92602  0.0000000  514.74950  54.767648
 9 |   0.0000000 -152.47720  106.34288  529.91606  64.487389
10 |   33.628919 -134.59895  111.38430  545.48293  64.606642
11 |   476.74609  101.04479  177.06432  751.27948  67.625345
```

Forward Stagewise:

```
optn = [ 3 , ., ., 3 ];
modl = "11 = 1:10";
< gof,parm,yptr > = lars(Diab,modl,optn);
```

```
                  ****************************
                  Iteration History: Stagewise
                  ****************************
```

Iter	Nact	Add	Dropp	gamma	arclen	R2
1	1	3	.	0.06332106	60.1192426	0.04217625
2	2	9	.	0.49073113	513.223759	0.35125417
3	3	4	.	0.30211221	175.553216	0.41733717
4	4	7	.	0.58828998	259.367461	0.47892806
5	5	2	.	0.31774527	88.6592445	0.49480593
6	6	10	.	0.22321055	43.6778396	0.50059988
7	7	5	.	0.71027043	135.984058	0.51341086
8	6	8	7	0.72612474	46.2667257	0.51484366
9	7	7	.	0.13604160	5.69939070	0.51501346
10	8	1	.	0.00159142	0.05836568	0.51501510
11	9	3	.	0.18745372	6.93264580	0.51519240
12	9	6	3	7.416e-016	6.067e-014	0.51519240
13	9	3	3	0.76204054	62.3447051	0.51760153
14	10	3	.	1.00000000	19.6893296	0.51774943

****
Iteration History: Stagewise
****

Iter	Nact	Add	Dropp	SSE	R2	Cp
0	0	.	.	2621009.12	0.00000000	445.431827
1	1	3	.	2510464.79	0.04217625	410.087634
2	2	9	.	1700368.75	0.35125417	138.420209
3	3	4	.	1527164.60	0.41733717	81.9082175
4	4	7	.	1365734.31	0.47892806	29.3736813
5	5	2	.	1324118.28	0.49480593	17.3149119
6	6	10	.	1308932.27	0.50059988	14.1847610
7	7	5	.	1275354.57	0.51341086	4.84151062
8	6	8	7	1271599.20	0.51484366	5.57286655
9	7	7	.	1271154.14	0.51501346	7.42251499
10	8	1	.	1271149.86	0.51501510	9.42106988
11	9	3	.	1270685.14	0.51519240	11.2640771
12	9	6	3	1270685.14	0.51519240	13.2640771
13	9	3	3	1264370.80	0.51760153	13.1309630
14	10	3	.	1263983.13	0.51774943	15.0000000

Number Updates= 14
Number Downdates= 7
Total Processing Time: 0

Parameter Estimates (Stagewise)
****

Dense Matrix (15 by 10)

	1	2	3	4	5
1	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2	0.0000000	0.0000000	60.119242	0.0000000	0.0000000
3	0.0000000	0.0000000	361.89461	0.0000000	0.0000000
4	0.0000000	0.0000000	434.75795	79.236444	0.0000000
5	0.0000000	0.0000000	505.65955	191.26988	0.0000000
6	0.0000000	-74.916586	511.34807	234.15466	0.0000000
7	0.0000000	-111.97854	512.04408	252.52702	0.0000000
8	0.0000000	-197.75647	522.26484	297.15973	-103.94623
9	0.0000000	-229.78472	522.26484	313.41210	-148.45549
10	0.0000000	-230.85773	522.26484	314.63528	-159.36762
11	-0.0103768	-230.86749	522.26484	314.64979	-159.47633
12	-1.2286691	-231.86285	523.45552	316.07390	-172.42413
13	-1.2286691	-231.86285	523.45552	316.07390	-172.42413
14	-7.9076880	-237.56378	523.45552	321.75792	-643.52167
15	-10.012180	-239.81908	519.83978	324.39043	-792.18454

	6	7	8	9	10
1	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
3	0.0000000	0.0000000	0.0000000	301.77537	0.0000000
4	0.0000000	0.0000000	0.0000000	374.91586	0.0000000
5	0.0000000	-114.10098	0.0000000	439.66496	0.0000000
6	0.0000000	-169.71144	0.0000000	450.66748	0.0000000
7	0.0000000	-196.04544	0.0000000	452.39276	12.078126
8	0.0000000	-223.92602	0.0000000	514.74950	54.767648
9	0.0000000	-223.92602	34.917346	524.22305	65.124670
10	0.0000000	-210.83324	50.022100	525.90522	65.669083

	6	7	8	9	10
11	0.0000000	-210.70070	50.174191	525.92295	65.676101
12	0.0000000	-194.70369	68.164735	527.83118	66.319994
13	3.52e-013	-194.70369	68.164735	527.83118	66.319994
14	361.98269	30.983071	151.30419	697.10785	66.902685
15	476.74609	101.04479	177.06432	751.27948	67.625345

2. NIR Spectra Data Set: nobs=21, nvar=268

```
  print "NIR Spectra data set: train: nr=21, test: nr=7";
  options NOECHO;
#include "..\\tdata\\nir.dat"
  options ECHO;
  nrtrn = nrow(xtrn); nctrn = ncol(xtrn);
  print "nrtrn,nctrn=",nrtrn,nctrn;
  xytrn = xtrn -> ytrn‘; /* attrib(xytrn); */
  /* free xtrn,ytrn; */
  nrtst = nrow(xtst); nctst = ncol(xtst);
  print "nrtst,nctst=",nrtst,nctst;
  xytst = xtst -> ytst‘; /* attrib(xytst); */
  /* model: */
  xind = [ 1:268 ]; yind = 269;
```

LARS:

```
  optn = [ 3 , ., ., 1 ];
  modl = "269 = 1:268";
  < gof,parm,yptr > = lars(xytrn,modl,optn);
```

```
                    **********************
                    Iteration History: LARS
                    **********************

   Iter   Nact    Add  Dropp       gamma      arclen         SSE

      1      1     40      .  0.17551292  23.5090643  12929.1301
      2      2     79      .  0.00834908  0.94111448  12717.8497
      3      3     80      .  0.16137627  18.0597683  9001.83404
      4      4     51      .  0.84948972  80.4847794  228.596488
      5      5     39      .  0.86547782  12.8622479  11.7308223
      6      6     65      .  0.24017961  0.64773258  8.65668197
      7      7     59      .  0.01583359  0.03742730  8.48114213
      8      8      1      .  0.47116378  1.25699678  3.35421208
      9      9      2      .  0.16251421  0.25764332  2.60367799
     10     10     64      .  0.36095034  0.48411178  1.53944685
     11     11     78      .  0.24811932  0.22531673  1.18099499
     12     12     63      .  0.11826313  0.09094567  1.04938964
     13     13     66      .  0.22838122  0.15713648  0.85784768
     14     14     50      .  0.17478760  0.10429325  0.74426420
     15     15    145      .  0.74376966  0.57958811  0.17688978
     16     16     67      .  0.21124133  0.04520162  0.15958839
     17     17    231      .  0.15307434  0.04365391  0.13659551
```

18	18	245	.	0.10114032	0.03215293	0.11718623
19	19	68	.	0.44149017	0.14345977	0.04453403
20	20	60	.	1.00000000	0.21103087	2.186e-019
21	21	17	.	0.37604055	1.010e-005	2.953e-019

****
Iteration History: LARS
****

Iter	Nact	Add	Dropp	R2	Cp	BIC
0	0	.	.	0.00000000	-19.0000000	144.597995
1	1	40	.	0.30765090	-17.0000000	138.972031
2	2	79	.	0.31896488	-15.0000000	140.929918
3	3	80	.	0.51795584	-13.0000000	136.218329
4	4	51	.	0.98775876	-11.0000000	61.9077812
5	5	39	.	0.99937182	-9.00000000	2.70133126
6	6	65	.	0.99953644	-7.00000000	-0.13000959
7	7	59	.	0.99954584	-5.00000000	3.45977645
8	8	1	.	0.99982038	-3.00000000	-11.4316590
9	9	2	.	0.99986057	-1.00000000	-11.4645450
10	10	64	.	0.99991756	1.00000000	-16.3461591
11	11	78	.	0.99993676	3.00000000	-14.6614680
12	12	63	.	0.99994381	5.00000000	-8.47814404
13	13	66	.	0.99995406	7.00000000	-2.18612437
14	14	50	.	0.99996014	9.00000000	7.86248557
15	15	145	.	0.99999053	11.0000000	-5.81176894
16	16	67	.	0.99999145	13.0000000	13.4467247
17	17	231	.	0.99999269	15.0000000	38.6346740
18	18	245	.	0.99999372	17.0000000	73.0412195
19	19	68	.	0.99999762	19.0000000	91.2234923
20	20	60	.	1.00000000	21.0000000	-924.243978
21	21	17	.	1.00000000	23.0000000	.

The stepwise parameter matrix ist stored in sparse format:

Parameter Estimates (LARS)
****

Sparse Matrix (22 by 268)

```
2 |          40
     -60.400954

3 |          40         79
```

```
      -61.634952  0.3971953

  4 |         40         79         80
      -86.548640 -77.411818  83.295238

  5 |         40         51         79         80
       237.32356 -301.51929 -420.21323  461.15773

  6 |         39         40         51         79         80
      -523.72095  604.37223 -177.06100  126.10677 -52.504544

  7 |         39         40         51         65         79
      -507.17133  561.77183 -159.59175  9.2629661  150.54234
    |         80
      -77.795757


  8 |         39         40         51         59         65
      -505.75010  559.48823 -159.22269  0.6211567  9.3200042
    |         79         80
       151.59333 -78.972441

  9 |          1         39         40         51         59
       5.5452875 -540.81531  590.10054 -162.78437  15.241256
    |         65         79         80
       1.6042095  243.21663 -170.85665

 10 |          1          2         39         40         51
       3.0210880  3.0091993 -572.97840  624.80452 -166.15671
    |         59         65         79         80
       20.555247 -2.7294927  281.46191 -208.67811

 11 |          1          2         39         40         51
      -1.1722001  8.1448713 -612.28689  669.49097 -172.66265
    |         59         64         65         79         80
       30.940806  4.6785395 -15.532455  348.77771 -275.70442

 12 |          1          2         39         40         51
      -2.5603526  10.087314 -621.18365  680.50980 -175.11066
    |         59         64         65         78         79
       34.867221  8.1203379 -21.144746  42.972158  299.53989
    |         80
      -269.65699

 13 |          1          2         39         40         51
      -3.2337433  10.566855 -623.93583  682.52617 -174.55006
```

```
    |          59          63          64          65          78
        35.967539   4.0301902   2.9373615  -20.658338   59.735756
    |          79          80
        276.72236  -263.37486

 14 |           1           2          39          40          51
       -4.3480272   11.252960  -634.13099   690.76416  -173.32020
    |          59          63          64          65          66
        37.589537   9.2832311  -2.4221604  -22.562693   1.7296107
    |          78          79          80
        93.050666   229.53836  -249.30183

 15 |           1           2          39          40          50
       -5.4457549   12.039458  -589.87398   647.66590  -20.028440
    |          51          59          63          64          65
       -153.34866   38.506056   12.725702  -6.1504418  -24.110063
    |          66          78          79          80
        2.1929149   111.64912   207.58743  -245.51926

 16 |           1           2          39          40          50
       -4.1563845   8.4315193  -389.27180   379.83243  -180.31196
    |          51          59          63          64          65
        69.905980   15.594580   35.435550  -12.188117  -37.320961
    |          66          78          79          80         145
        6.5968350   153.10314   156.45895  -240.94487   33.979886


 17 |           1           2          39          40          50
       -4.1159853   8.5770878  -389.85023   380.17204  -185.80942
    |          51          59          63          64          65
        75.604028   14.300578   35.645148  -12.554303  -36.134714
    |          66          67          78          79          80
        3.9465781   2.5476040   159.69349   147.86562  -239.24883
    |         145
        34.804280

 18 |           1           2          39          40          50
       -3.7622233   8.6792321  -424.82136   433.49333  -164.95304
    |          51          59          63          64          65
        46.718545   13.588598   32.876709  -14.739298  -24.811975
    |          66          67          78          79          80
       -4.4792586   9.1353422   159.23995   131.34776  -223.79561
    |         145         231
        35.502034  -15.414865

 19 |           1           2          39          40          50
```

```
       -3.5771872  8.4399768 -404.02226  405.76769 -159.97909
   |          51         59         63         64         65
        51.836499  11.864989  29.270482 -9.1005249 -28.540471
   |          66         67         78         79         80
       -2.1236049  8.5596952  158.51705  123.33814 -216.82745
   |         145        231        245
        35.776972 -30.017553  15.330770

20 |           1          2         39         40         50
       -1.2067826  4.8883937 -464.45859  414.48864 -163.40514
   |          51         59         63         64         65
        92.465486  5.0157186  12.325629  20.675325 -55.048233
   |          66         67         68         78         79
        14.385602 -1.3094794  7.7195489  154.75155  141.23563
   |          80        145        231        245
       -235.24320  35.789710 -44.888765  59.783604

21 |           1          2         39         40         50
       -1.0115705  2.7078262 -1269.3488  1299.7593 -126.28264
   |          51         59         60         63         64
       -66.486302 -30.425397  43.772299 -12.021143  6.5437740
   |          65         66         67         68         78
        3.8195358 -49.879293  48.003815  33.013994  263.38558
   |          79         80        145        231        245
       -4.3281288 -191.94870  28.798718  85.247308 -81.692302

22 |           1          2         17         39         40
       -8.2842069  13.474412  1.3968578 -3289.5315  3697.7851
   |          50         51         59         60         63
       -99.341784 -570.60702 -44.432171  113.22193 -27.400861
   |          64         65         66         67         68
       -111.95806  224.96204 -262.57924  198.76047  72.739541
   |          78         79         80        145        231
        501.53573 -207.39964 -194.77809  4.7958802  508.12317
   |         245
       -586.12399


                  Number Updates= 21
                  Number Downdates= 0
               Total Processing Time: 0
```

LASSO:

```
optn = [ 3 , ., ., 2 ];
modl = "269 = 1:268";
```

```
< gof,parm,yptr > = lars(xytrn,modl,optn);
```

```
                 ************************
                  Iteration History: LASSO
                  ************************
```

Iter	Nact	Add	Dropp	gamma	arclen	SSE
1	1	40	.	0.17551292	23.5090643	12929.1301
2	2	79	.	0.00834908	0.94111448	12717.8497
3	2	80	.	8.238e-004	0.09219055	12697.2240
4	2	.	79	0.16065205	17.9464235	9009.71197
5	2	51	.	0.21991671	20.8169688	5502.05530
6	2	.	40	0.38966002	28.6207376	2116.78482
7	2	81	.	0.08267275	3.72064078	1795.73729
8	2	.	80	0.30115007	12.3888452	929.905155
9	3	50	.	0.31014952	9.37798047	450.728324
10	3	80	.	0.13578764	2.86050725	338.391735
11	2	.	81	0.28663226	5.17269814	178.450745
12	2	.	51	0.23730439	2.94051493	114.223642
13	3	64	.	0.29761451	3.07401447	60.1711616
14	3	79	.	0.30787884	2.29674478	31.1792337
15	3	.	80	0.31184129	1.56742058	17.8792610
16	3	78	.	0.34305648	1.26518363	10.1480200
17	3	.	79	0.04671903	0.10629664	9.67561996
18	4	244	.	0.31311277	0.71272542	6.93890369
19	5	60	.	0.11603106	0.22933040	6.08497282
20	6	65	.	0.15471575	0.27052636	5.21210616
21	6	245	.	0.00585121	0.01043938	5.17496450
22	6	.	244	0.17205506	0.25310603	4.49435087
23	7	77	.	0.09170360	0.12454628	4.17156024
24	7	51	.	0.03294796	0.04256447	4.06339647
25	7	.	65	0.20629611	0.25030387	3.51864947
26	7	63	.	0.09987137	0.14496895	3.11880417
27	7	.	64	0.05658037	0.04894022	3.03653584
28	7	1	.	0.02792156	0.03040063	2.97126042
29	7	.	78	0.30533830	0.25206634	2.61862052
30	7	145	.	0.06330884	0.05600588	2.52266649
31	7	.	51	0.16249610	0.10378517	2.40086408
32	8	20	.	0.14297359	0.10932369	2.24562865
33	9	231	.	0.43111534	0.59621071	0.95203732
34	10	51	.	0.08832440	0.07164010	0.84095475
35	11	66	.	0.36498753	0.27617992	0.49926871
36	11	19	.	0.05427539	0.02861475	0.46991531
37	10	.	20	0.06787927	0.03105781	0.44245921

38	9	.	50	0.13310898	0.05129279	0.40555939
39	9	.	60	0.16753160	0.04765827	0.38071569
40	10	57	.	0.08574639	0.02935564	0.36147739
41	11	20	.	0.00345445	0.00117307	0.36068206
42	12	158	.	0.14841991	0.05294976	0.32570539
43	12	58	.	0.02259759	0.00815913	0.31988007
44	11	.	19	0.11469896	0.04018694	0.29333456
45	11	.	57	0.04745496	0.01356363	0.28576499
46	12	232	.	0.06627219	0.02451542	0.26822849
47	12	57	.	0.07121752	0.02641440	0.24933214
48	12	.	158	0.08378224	0.02731934	0.23226215
49	13	30	.	0.09924487	0.04077885	0.20041373
50	13	158	.	0.10060761	0.03905952	0.17161072
51	13	.	58	0.06310839	0.02186364	0.15693960
52	14	165	.	0.08614858	0.03047732	0.13630416
53	15	58	.	0.16475268	0.05355962	0.10434930
54	16	62	.	0.02617899	0.00737977	0.10024309
55	16	19	.	0.01840394	0.00562121	0.09684086
56	15	.	20	0.09331534	0.02497606	0.08409487

****
Iteration History: LASSO
****

Iter	Nact	Add	Dropp	R2	Cp	BIC
0	0	.	.	0.00000000	-7994259.16	144.597995
1	1	40	.	0.30765090	-5534821.94	138.972031
2	2	79	.	0.31896488	-5444373.34	140.929918
3	2	80	.	0.32006938	-5435541.70	140.895833
4	2	.	79	0.51753398	-3856960.21	133.691147
5	2	51	.	0.70536742	-2355372.63	123.334338
6	2	.	40	0.88664713	-906177.096	103.274647
7	2	81	.	0.90383908	-768738.512	99.8205110
8	2	.	80	0.95020400	-398084.200	86.0006553
9	3	50	.	0.97586370	-192952.369	73.3376335
10	3	80	.	0.98187928	-144860.427	67.3177615
11	2	.	81	0.99044404	-76389.6152	51.3344859
12	2	.	51	0.99388338	-48892.7659	41.9652451
13	3	64	.	0.99677786	-25751.5498	31.0505303
14	3	79	.	0.99833037	-13338.4545	17.2442713
15	3	.	80	0.99904257	-7642.89635	5.56594342
16	3	78	.	0.99945658	-4331.24516	-6.32767595
17	3	.	79	0.99948187	-4127.01641	-7.32873112

18	4	244	.	0.99962842	-2953.46112	-11.4833253
19	5	60	.	0.99967415	-2585.90360	-11.0830162
20	6	65	.	0.99972089	-2210.23989	-10.7843066
21	6	245	.	0.99972288	-2192.34000	-10.9344891
22	6	.	244	0.99975933	-1898.97749	-13.8957250
23	7	77	.	0.99977661	-1758.79471	-11.4408786
24	7	51	.	0.99978241	-1710.49110	-11.9925681
25	7	.	65	0.99981158	-1475.29147	-15.0153491
26	7	63	.	0.99983299	-1302.12251	-17.5485285
27	7	.	64	0.99983739	-1264.90443	-18.1099071
28	7	1	.	0.99984089	-1234.96081	-18.5662600
29	7	.	78	0.99985977	-1081.99991	-21.2193703
30	7	145	.	0.99986491	-1038.92315	-22.0033252
31	7	.	51	0.99987143	-984.781005	-23.0425684
32	8	20	.	0.99987975	-916.326585	-19.8575183
33	9	231	.	0.99994902	-360.555715	-32.5921430
34	10	51	.	0.99995497	-311.002605	-29.0436114
35	11	66	.	0.99997326	-162.730922	-32.7417988
36	11	19	.	0.99997484	-148.165085	-34.0142300
37	10	.	20	0.99997631	-134.411460	-42.5295926
38	9	.	50	0.99997828	-116.615093	-50.5122181
39	9	.	60	0.99997961	-103.979802	-51.8397218
40	10	57	.	0.99998064	-93.7441169	-46.7747172
41	11	20	.	0.99998069	-91.4036460	-39.5698984
42	12	158	.	0.99998256	-74.4305578	-33.0475288
43	12	58	.	0.99998287	-69.9368063	-33.4265186
44	11	.	19	0.99998429	-56.5729927	-43.9102423
45	11	.	57	0.99998470	-51.3325498	-44.4592674
46	12	232	.	0.99998564	-41.8253868	-37.1247646
47	12	57	.	0.99998665	-31.7360867	-38.6588837
48	12	.	158	0.99998756	-22.4286294	-40.1481872
49	13	30	.	0.99998927	-6.79470284	-32.7210210
50	13	158	.	0.99999081	7.53551639	-35.9792801
51	13	.	58	0.99999160	15.8160467	-37.8560006
52	14	165	.	0.99999270	26.6498291	-27.7851656
53	15	58	.	0.99999441	42.3293231	-16.8952106
54	16	62	.	0.99999463	46.0871414	3.68172817
55	16	19	.	0.99999481	49.5435943	2.95661700
56	15	.	20	0.99999550	57.0000000	-21.4269759

Parameter Estimates (LASSO)

****

Sparse Matrix (57 by 268)

```
 2 |          40
      -60.400954

 3 |          40         79
      -61.634952  0.3971953

 4 |          40         80
      -61.762130  0.4252011

 5 |          40         80
      -85.298367  7.8444953

 6 |          51         80
      -78.076558  19.172062

 7 |          51         80
      -106.24218  31.053960

 8 |          51         81
      -110.44585  31.546648

 9 |          51         81
      -122.64210  36.559024

10 |          50         51         81
      -60.808394 -61.645369  45.636882

11 |          50         51         80
      -85.740620 -34.785486  50.611087

12 |          50         80
      -120.15282  55.884445

13 |          50         80
      -123.21128  57.122050

14 |          50         64         80
      -123.01704  8.0694546  58.587749

15 |          50         64         79
      -121.74278  16.220087  61.082285

16 |          50         64         79
      -121.61105  20.364277  61.855003
```

```
17 |          50          64          78
      -120.31832   26.212239   63.391660

18 |          50          64          78
      -120.30681   26.495665   63.445502

19 |          50          64          78         244
      -121.57251   25.284169   63.346150   8.0878742

20 |          50          60          64          78         244
      -121.98521   4.1014173   21.487589   63.494412   8.8660243

21 |          50          60          64          65          78
      -122.50920   8.9035473   16.702948   0.5968751   63.662908
   |         244
       9.5087266

22 |          50          60          64          65          78
      -122.38595   9.0232939   16.472749   0.6971427   63.753962
   |         245
       9.3470720

23 |          50          60          64          65          78
      -122.87802   13.487955   12.039110   1.1168594   63.911140
   |         245
       10.153654

24 |          50          60          64          65          77
      -122.66394   15.924147   11.120769   0.7749059   14.009576
   |          78         245
       50.110152   9.1203795

25 |          50          51          60          64          77
      -121.18913  -1.4803467   16.940288   11.307860   16.950286
   |          78         245
       47.123555   8.9755569

26 |          50          51          60          64          77
      -117.17701  -5.5041782   22.162089   9.4256034   38.244365
   |          78         245
       25.876618   6.5719790

27 |          50          51          60          63          77
      -118.72326  -3.4639029   23.239716   6.8286605   52.569980
   |          78         245
       12.218253   7.8009042
```

```
28 |          50          51          60          63          77
     -117.59243 -4.4760320  23.874316  7.0849524  58.639267
   |          78         245
     6.1701355  6.9568457

29 |           1          50          51          60          63
     0.2067151 -118.68273 -3.4273368  23.758783  7.0779844
   |          77         245
     64.842202  7.1958515

30 |           1          50          51          60          63
     1.3367087 -116.85253 -6.9721463  24.425304  6.2684775
   |          77         245
     64.046150  10.384475

31 |           1          50          60          63          77
     1.1512668 -123.18661  24.137744  7.2509338  63.474112
   |         145         245
     2.8227366  7.1426546

32 |           1          50          60          63          77
     1.1622385 -123.21284  24.980973  7.9846878  62.797917
   |         145         245
     4.5100338  4.9394188

33 |           1          20          50          60          63
     0.9070923 -0.2038649 -121.22820  24.274252  9.6250288
   |          77         145         245
     62.461659  6.9097086  0.7123958

34 |           1          20          50          60          63
     1.3603080 -0.4106777 -69.050551  10.783934  10.820136
   |          77         145         231         245
     51.446028  25.604348 -125.86572  83.897684

35 |           1          20          50          51          60
     1.5032760 -0.4940418 -56.578249 -5.8656954  9.2287008
   |          63          77         145         231         245
     10.696459  50.357564  26.474316 -141.76101  95.669126

36 |           1          20          50          51          60
     1.6830999 -0.7645294 -12.884532 -23.306737  2.8579728
   |          63          66          77         145         231
     9.1684591  2.3392118  46.035247  29.638387 -206.69379
   |         245
```

```
       143.05189

37 |          1          19          50          51          60
       1.8537366 -0.6458209 -5.2899052 -26.696994  1.9402235
   |         63          66          77         145         231
       9.1072957  2.4428427  45.118344  29.676559 -219.49856
   |        245
       153.33077

38 |          1          19          51          60          63
       1.8828592 -0.6736706 -28.960886  1.2108562  8.9407087
   |         66          77         145         231         245
       2.6931078  44.625283  29.972737 -227.04432  158.90724

39 |          1          19          51          63          66
       1.7826355 -0.6555324 -24.783064  8.8915400  3.3079143
   |         77         145         231         245
       43.602690  32.106964 -237.51930  165.07796

40 |          1          19          51          63          66
       1.6133804 -0.6298097 -21.661499  8.7053341  3.7407656
   |         77         145         231         245
       42.732544  34.045809 -245.31507  168.61694

41 |          1          19          51          57          63
       1.5468006 -0.6310247 -19.617133 -0.7086843  8.7873109
   |         66          77         145         231         245
       4.0914506  42.339935  35.334199 -248.94856  169.64764
                                                          Page 14

42 |          1          19          20          51          57
       1.5415640 -0.6110881 -0.0256691 -19.565621 -0.7419818
   |         63          66          77         145         231
       8.7874059  4.1095558  42.341724  35.378796 -248.92996
   |        245
       169.55878

43 |          1          19          20          51          57
       1.3326684 -0.0464764 -0.8003717 -16.179555 -1.8965525
   |         63          66          77         145         158
       8.9560630  4.4115993  40.637393  34.858296  11.480089
   |        231         245
      -253.96185  163.25190

44 |          1          20          51          57          58
       1.2893285 -0.8668883 -15.654538 -1.6050160 -0.5427418
```

```
    |          63          66          77         145         158
        8.9246021   4.5144549   40.665307   35.218651   11.308555
    |         231         245
       -254.40300   163.19167

45  |           1          20          51          58          63
        1.1004343  -0.9063014  -12.684524  -3.3111728   8.8107309
    |          66          77         145         158         231
        4.9326859   40.495313   36.840455   11.426128  -258.64236
    |         245
        163.95626

46  |           1          20          51          58          63
        1.0558811  -0.9211936  -11.614337  -3.6685919   8.8518981
    |          66          77         145         158         231
        5.0208186   40.127769   36.997328   13.313392  -260.65390
    |         245
        163.52767

47  |           1          20          51          58          63
        1.1150991  -0.9479357  -12.010594  -4.2977954   8.5787602
    |          66          77         145         158         231
        5.1283729   41.892813   38.613428   7.3659525  -235.81262
    |         232         245
       -22.843748   165.42040

48  |           1          20          51          57          58
        1.3365816  -0.9814686  -13.879134  -1.6595893  -3.1956175
    |          63          66          77         145         231
        8.2826088   5.0450765   44.562080   40.099913  -190.69509
    |         232         245
       -63.602775   167.24222

49  |           1          20          51          57          58
        1.5081848  -1.0322073  -14.791693  -3.2031983  -2.0882469
    |          63          66          77         145         231
        8.1233293   4.8020978   45.897961   40.157892  -155.34272
    |         232         245
       -97.687581   166.59152

50  |           1          20          30          51          57
        1.7169916  -0.8869292  -7.2110218  -15.482314  -4.9435176
    |          58          63          66          77         145
       -0.9039179   7.0899515   5.0737013   48.235100   41.125071
    |         231         232         245
       -107.44181  -133.73848   160.87296
```

```
51 |          1          20          30          51          57
       1.7930156 -0.6978552 -16.718527 -14.006030 -6.0847024
   |         63          66          77         145         158
       6.0864663  5.2714798  47.949394  39.620722  14.152216
   |        231         232         245
      -84.359357 -147.89983  148.97504

52 |          1          20          30          51          57
       1.7902416 -0.5880501 -22.140046 -12.745718 -6.2636901
   |         63          66          77         145         158
       5.5201821  5.4492464  47.565126  38.857236  22.404306
   |        231         232         245
      -76.823333 -150.94308  142.27191

53 |          1          20          30          51          57
       1.9805744 -0.4937958 -28.841055 -12.049166 -6.5440128
   |         63          66          77         145         158
       4.9702369  5.3525511  48.394423  38.468543  23.447765
   |        165         231         232         245
       6.7097384 -54.505416 -164.87133  132.83093

54 |          1          20          30          51          57
       2.2585405 -0.3345112 -40.624778 -10.068331 -5.9395203
   |         58          63          66          77         145
      -1.2164426  4.0575192  5.2507638  49.681203  38.156971
   |        158         165         231         232         245
       23.249201  20.313717 -25.039705 -180.22344  116.11035

55 |          1          20          30          51          57
       2.2747546 -0.3202449 -41.997012 -9.8644541 -5.7832579
   |         58          62          63          66          77
      -1.5065442  0.2042972  3.7313766  5.2869425  50.003562
   |        145         158         165         231         232
       38.270784  22.415584  22.065154 -20.788492 -182.43362
   |        245
       114.15517

56 |          1          19          30          51          57
       2.2920342 -0.2623542 -42.331718 -9.7697583 -5.5427091
   |         58          62          63          66          77
      -1.8706696  0.8283361  3.0582166  5.3012427  50.483222
   |        145         158         165         231         232
       38.614867  20.292141  22.983545 -15.253802 -188.09643
   |        245
       115.00904
```

```
57 |          1         19         30         51         57
       2.3517321 -0.2308397 -46.740942 -9.1975222 -5.0094388
   |         58         62         63         66         77
      -2.8826060  1.5018959  1.9984666  5.4114977  51.701059
   |        145        158        165        232        245
       39.092924  16.603645  29.218021 -196.25216  108.44880

                   Number Updates= 36
                  Number Downdates= 21
                 Total Processing Time: 0
```

Forward stagewise:

```
optn = [ 3 , ., ., 3 ];
modl = "269 = 1:268";
< gof,parm,yptr > = lars(xytrn,modl,optn);
```

```
                 ***************************
                  Iteration History: Stagewise
                  ***************************

  Iter   Nact    Add  Dropp       gamma      arclen         SSE

     1      1     40      .  0.17551292  23.5090643  12929.1301
     2      2     79      .  0.00834908  0.94111448  12717.8497
     3      2     80     79  1.020e-012  1.140e-010  12717.8497
     4      2     79     79  0.16134333  18.0385055  9009.75563
     5      2     51     40  0.54488296  51.3047195  1980.50014
     6      2     81     80  1.453e-012  6.228e-011  1980.50014
     7      2     80     80  0.22528877  9.65769490  1245.75770
     8      2     50     51  2.916e-014  9.816e-013  1245.75770
     9      2     51     51  0.31948948  10.7563456  637.182629
    10      2     64     50  9.548e-014  2.389e-012  637.182629
    11      2     50     50  0.54808364  13.7109122  139.184789
    12      2     80     81  0.38260429  4.32767779  60.0119360
    13      3    244      .  0.16376525  1.16859728  44.6997852
    14      3     79     80  0.02853902  0.17044319  42.6929653
    15      3     50     64  0.29357800  1.74117907  25.0711840
    16      3     78     79  1.286e-013  5.388e-013  25.0711840
    17      3     79     79  0.47194065  1.97758959  12.4085174
    18      3     77     78  0.20447755  0.45254277  10.6102078
    19      3    245    244  0.68864413  1.21778043  7.78621433
    20      3     19     50  4.858e-015  2.861e-015  7.78621433
    21      3     50     50  0.04184470  0.02464254  7.75779737
    22      3     76     77  2.416e-013  1.365e-013  7.75779737
```

23	3	77	77	0.37355537	0.21106450	7.56383625
24	3	144	76	0.01379212	0.00543083	7.55958883
25	3	59	245	3.080e-014	1.240e-014	7.55958883
26	3	245	245	0.34546921	0.13914574	7.46686202
27	3	3	19	1.740e-015	1.223e-015	7.46686202
28	3	19	19	0.09182469	0.06453835	7.38030655
29	4	65	.	0.09786862	0.11926548	7.10385021
30	5	20	.	0.03772645	0.04695375	6.98917904
31	5	60	20	0.04614871	0.06820439	6.79222871
32	5	96	3	0.15672324	0.27155293	5.92493511
33	6	246	.	0.08522456	0.12655865	5.56507245
34	6	1	59	3.792e-015	5.640e-015	5.56507245
35	6	59	59	0.17787496	0.26458923	4.84792624
36	7	21	.	0.01325644	0.01684685	4.80539064
37	7	245	246	0.03539703	0.04475692	4.69421019
38	7	76	144	0.00829705	0.01185169	4.66049226
39	7	22	21	0.06770102	0.09624930	4.39608446
40	7	77	76	0.02584723	0.03440627	4.30566914
41	7	23	22	0.09345173	0.12163587	4.00382426
42	7	3	60	0.02822173	0.03519199	3.91729518
43	7	246	245	0.03608410	0.04378924	3.81293332
44	7	225	23	0.09124999	0.11147194	3.55300873
45	8	59	.	0.02965211	0.03353994	3.47825862
46	7	32	246	0.22216006	0.28523729	2.82717124
47	6	55	225	0.04685138	0.05404493	2.70540614
48	7	251	.	0.05682241	0.07020622	2.53685023
49	7	225	55	1.585e-014	1.931e-014	2.53685023
50	7	55	55	0.24544833	0.29909219	1.89738604
51	7	97	96	3.355e-013	3.141e-013	1.89738604
52	7	96	96	0.05072090	0.04748735	1.81072118
53	8	67	.	0.19191605	0.17471661	1.52312992
54	9	20	.	0.08977036	0.06789749	1.42503194
55	8	66	67	6.037e-015	4.268e-015	1.42503194
56	8	67	67	0.01827399	0.01291951	1.40693094

****
Iteration History: Stagewise
****

Iter	Nact	Add	Dropp	R2	Cp	BIC
0	0	.	.	0.00000000	-477849.533	144.597995
1	1	40	.	0.30765090	-330842.537	138.972031
2	2	79	.	0.31896488	-325434.376	140.929918

3	2	80	79	0.31896488	-325432.376	140.929918
4	2	79	79	0.51753164	-230549.112	133.691249
5	2	51	40	0.89394511	-50685.2647	101.877119
6	2	81	80	0.89394511	-50683.2647	101.877119
7	2	80	80	0.93329024	-31880.9620	92.1414044
8	2	50	51	0.93329024	-31878.9620	92.1414044
9	2	51	51	0.96587916	-16304.9805	78.0621034
10	2	64	50	0.96587916	-16302.9805	78.0621034
11	2	50	50	0.99254672	-3558.40606	46.1157725
12	2	80	81	0.99678639	-1530.56200	28.4493338
13	3	244	.	0.99760635	-1136.76066	24.8088159
14	3	79	80	0.99771381	-1083.41094	23.8441906
15	3	50	64	0.99865745	-630.511673	12.6655751
16	3	78	79	0.99865745	-628.511673	12.6655751
17	3	79	79	0.99933553	-302.504303	-2.10448108
18	3	77	78	0.99943183	-254.489858	-5.39237940
19	3	245	244	0.99958305	-180.230614	-11.8910764
20	3	19	50	0.99958305	-178.230614	-11.8910764
21	3	50	50	0.99958457	-175.503492	-11.9678593
22	3	76	77	0.99958457	-173.503492	-11.9678593
23	3	77	77	0.99959496	-166.540491	-12.4995782
24	3	144	76	0.99959519	-164.431809	-12.5113739
25	3	59	245	0.99959519	-162.431809	-12.5113739
26	3	245	245	0.99960015	-158.059152	-12.7705553
27	3	3	19	0.99960015	-156.059152	-12.7705553
28	3	19	19	0.99960479	-151.844404	-13.0154081
29	4	65	.	0.99961959	-142.770547	-10.9899696
30	5	20	.	0.99962573	-137.836386	-8.17365856
31	5	60	20	0.99963628	-130.796898	-8.77392212
32	5	96	3	0.99968272	-106.604928	-11.6427193
33	6	246	.	0.99970199	-95.3969023	-9.40826113
34	6	1	59	0.99970199	-93.3969023	-9.40826113
35	6	59	59	0.99974040	-73.0468450	-12.3053995
36	7	21	.	0.99974267	-69.9584607	-8.47046609
37	7	245	246	0.99974863	-65.1136187	-8.96204382
38	7	76	144	0.99975043	-62.2508575	-9.11342856
39	7	22	21	0.99976459	-53.4852939	-10.3399720
40	7	77	76	0.99976943	-49.1717820	-10.7763874
41	7	23	22	0.99978560	-39.4482928	-12.3027219
42	7	3	60	0.99979023	-35.2342205	-12.7615416
43	7	246	245	0.99979582	-30.5638501	-13.3285965
44	7	225	23	0.99980974	-21.9130011	-14.8112809
45	8	59	.	0.99981374	-18.0003244	-10.6690466
46	7	32	246	0.99984861	0.65944543	-19.6101615

```
47      6     55    225  0.99985513  5.77512326 -24.5546782
48      7    251      .  0.99986415  12.0880660 -21.8855830
49      7    225     55  0.99986415  14.0880660 -21.8855830
50      7     55     55  0.99989840  32.4504264 -27.9849506
51      7     97     96  0.99989840  34.4504264 -27.9849506
52      7     96     96  0.99990304  38.6679729 -28.9667418
53      8     67      .  0.99991844  48.0267458 -28.0100989
54      9     20      .  0.99992369  52.5368389 -24.1218925
55      8     66     67  0.99992369  54.5368389 -29.4081351
56      8     67     67  0.99992466  57.0000000 -29.6765892

                Number Updates= 56
              Number Downdates= 93
            Total Processing Time: 0
```

3.4 Function `nlkpca`

< gof,pc,eval > = nlkpca(x,optn<,class<,kfun>>)

Purpose: The `nlkpca` function implements a (nonlinear) Hilbert kernel method for principal components analysis.

Input: **x** the $N \times M$ input data set $\mathbf{X}$ for which the $N \times p$ matrix $\mathbf{P}$ of principal components is computed. The number of components (factors) p is either user specified or the rank of $\mathbf{X}$.

optn : is specified in form of a two column matrix:

Option Name	Second Column	Meaning
"fact"	int	number p of factors (components)
"kern"		link function, default is linear
	"line"	linear function
	"poly"	polynomial function
	"rbf"	Gaussian radial basis function (def. scaling $[0, 1]$)
	"rbf2"	mod. Gaussian radial basis function (def. scaling $[0, 1]$)
	"erbf"	Exponential radial basis function
	"tanh"	sigmoid function (same as "sigm")
	"sigm"	sigmoid function (same as "tanh")
	"four"	Fourier function (def. scaling $[-\pi/2, \pi/2]$)
	"spli"	spline function (def. scaling $[0, 1]$)
	"anov"	anova function
	"curv"	curvspline function (def. scaling $[0, 1]$)
	"bspl"	Bspline function (def. scaling $[0, 1]$)
	"anob"	anova spline function (def. scaling $[0, 1]$)
	"ano1"	modified anova function (def. scaling $[0, 1]$)
	"ano2"	modified anova function (def. scaling $[0, 1]$)
	"ano3"	modified anova function (def. scaling $[0, 1]$)
"kfp1"	real	first parameter of kernel function
"kfp2"	real	second parameter of kernel function
"kfp3"	real	third parameter of kernel function
"nopr"		no printed output in ...txt (but warnings or errors are in ...log)
"pall"		is equivalent to "print" 10
"pmat"		print kernel matrix (this can be big)
"print"	int	amount of general printed output (default is 3)
"pshort"		less than default output
"psum"		much less than default output
"scale"	int	if not zero, perform scaling of data; this is default for several kernel functions

Some of the kernel functions, especially those based on the `exp` function, like all `rbf` functions, can create kernel matrices which are diagonal or even identity matrices for extrem parameter settings. Using the identity matrix for the kernel matrix produces useless results.

class : This optional argument should be an integer scalar or vector of integer scalars naming the number of columns which are considered categorical (nominal scaled) variables.

kfun This optional argument specifies the module name for a user specified kernel function. Note, the kernel matrix should satisfy the Mercer property of positive definiteness.

Output: **gof** vector of some goodness-of-fit measures.

pc $N \times p$ matrix $\mathbf{P}$ of principal components

eval matrix with eigenvalues of the centered kernel matrix; additional columns contain differences, proportions and cumulated values.

Restrictions: 1. The input data set should not contain missing values, string or complex data.

Relationships: nlkpls(), eig(), svm()

Examples: 1. Australian Institute of Sport (ais) data set (Weisberg, 2002):

```
options NOECHO;
ais = [
%inc "..\\tdata\\ais.dat";
];
options ECHO;
cname = [ "Sex" "Ht" "Wt" "LBM" "RCC" "WCC" "Hc" "Hg" ];
ais = shape(ais,.,8);
nobs = nrow(ais); nvar = ncol(ais);
ais = ais -> log(ais[,5]);
```

Using the "nlkpca" method with linear kernel:

```
optn = [ "kern"    "line" ,
         "fact"         6 ,
         "print"        2 ];
< gof,pc,eval > = nlkpca(ais,optn);
print "gof=" , gof;
```

```
                    *****************
                    Model Information
                    *****************

              Number Valid Observations  202
              Number Response Variables    0
              N Independend Variables      9
              Kernel Function         Linear
              Use Unscaled Data

         Memory needed for Kernel matrix: 0.156425 Mb


             Eigenvalues of Centered Kernel Matrix
             ************************************

 N  Eigenvalue  Difference  Proportion  Cumulative
 1  85027.4890  79274.2033  0.88406286  0.88406286
 2  5753.28568  2136.94956  0.05981908  0.94388194
```

```
          3  3616.33611  2445.45001  0.03760041  0.98148235
          4  1170.88610  605.476953  0.01217414  0.99365649
          5  565.409146  536.733574  0.00587877  0.99953526
          6  28.6755719  18.7134747  2.982e-004  0.99983342


   Kernel PCA Number of Factors . . . . . . . . . . . . . . . . . 6
   Variance Explained . . . . . . . . . . . . . . . . . . 0.999833
   Remaining Variance . . . . . . . . . . . . . . . . . 0.000166585
   Kernel Function. . . . . . . . . . . . . . . . . . . . . Linear
   Total Number of Kernel Calls . . . . . . . . . . . . . . 20908
   Total Processing Time. . . . . . . . . . . . . . . . . . . . 0


                Total Number of Kernel Calls: 20908
                     Time for Optimization: 0
                     Total Processing Time: 0
```

The following is the output of the `gof` return argument:

```
gof=
               |          1
---------------------------
   VarExplain |       1.00
    RemainVar |   1.7e-004
  KernelCalls |   20908.00
    TotalTime |       0.00
   TuningTime |       0.00
```

The following small examples with the RBF2 kernel show that the spectrum of eigenvalues of the kernel matrix become closer to one for increasing values of the parameter `kfp1`. That means, that more variance is explained by smaller values of the kernel function parameter.

```
    optn = [ "kern"    "rbf2" ,
             "kfp1"      .001 ,
             "fact"         6 ,
             "print"        2 ];
    < gof,pc,eval > = nlkpca(ais,optn);
    print "gof=" , gof;


                Eigenvalues of Centered Kernel Matrix
                *************************************
```

```
N  Eigenvalue  Difference  Proportion  Cumulative
1  0.56787328  0.49693513  0.74938665  0.74938665
2  0.07093815  0.01246475  0.09361261  0.84299925
3  0.05847340  0.01981441  0.07716367  0.92016292
4  0.03865899  0.02730555  0.05101584  0.97117876
5  0.01135344  0.00486527  0.01498241  0.98616117
6  0.00648817  0.00472845  0.00856203  0.99472320
```

```
Kernel PCA Number of Factors . . . . . 6
Variance Explained . . . . . 0.994723
Remaining Variance . . . . . 0.0052768
Kernel Function. . . . . RBF2
Kernel Parameter [1] . . . . . 0.001
Total Number of Kernel Calls . . . . . 20908
Total Processing Time. . . . . 1
```

Now, for `kfp1=.01` the first 6 eigenvalues explain less of the variance of the kernel matrix:

```
Eigenvalues of Centered Kernel Matrix
************************************
```

```
N  Eigenvalue  Difference  Proportion  Cumulative
1  48.7147731  35.4515394  0.31269864  0.31269864
2  13.2632338  1.39564658  0.08513629  0.39783494
3  11.8675872  2.38925071  0.07617768  0.47401262
4  9.47833648  1.69308999  0.06084115  0.53485377
5  7.78524649  0.54806431  0.04997326  0.58482703
6  7.23718218  0.16933449  0.04645525  0.63128228
```

```
Kernel PCA Number of Factors . . . . . 6
Variance Explained . . . . . 0.631282
Remaining Variance . . . . . 0.368718
Kernel Function. . . . . RBF2
Kernel Parameter [1] . . . . . 1
Total Number of Kernel Calls . . . . . 20908
Total Processing Time. . . . . 1
```

Now, for `kfp1=.1` the first 6 eigenvalues explain less of the variance of the kernel matrix:

```
Eigenvalues of Centered Kernel Matrix
************************************
```

```
N  Eigenvalue  Difference  Proportion  Cumulative
1  38.4773225  33.9002668  0.67433083  0.67433083
2  4.57705565  0.43882766  0.08021477  0.75454561
3  4.13822799  1.46787901  0.07252414  0.82706975
4  2.67034898  0.61003673  0.04679896  0.87386870
5  2.06031225  0.77429050  0.03610782  0.90997652
6  1.28602175  0.51708816  0.02253806  0.93251458
```

```
Kernel PCA Number of Factors . . . . . 6
Variance Explained . . . . . 0.932515
Remaining Variance . . . . . 0.0674854
Kernel Function. . . . . RBF2
Kernel Parameter [1] . . . . . 0.1
Total Number of Kernel Calls . . . . . 20908
Total Processing Time. . . . . 0
```

Now, for `kfp1=1.` the first 6 eigenvalues explain less of the variance of the kernel matrix:

```
Eigenvalues of Centered Kernel Matrix
*************************************
```

```
N  Eigenvalue  Difference  Proportion  Cumulative
1  48.7147731  35.4515394  0.31269864  0.31269864
2  13.2632338  1.39564658  0.08513629  0.39783494
3  11.8675872  2.38925071  0.07617768  0.47401262
4  9.47833648  1.69308999  0.06084115  0.53485377
5  7.78524649  0.54806431  0.04997326  0.58482703
6  7.23718218  0.16933449  0.04645525  0.63128228
```

```
Kernel PCA Number of Factors . . . . . 6
Variance Explained . . . . . 0.631282
Remaining Variance . . . . . 0.368718
Kernel Function. . . . . RBF2
Kernel Parameter [1] . . . . . 1
Total Number of Kernel Calls . . . . . 20908
Total Processing Time. . . . . 0
```

Now, for `kfp1=10` less than 13 % of the variance of the kernel matrix are explained by the first six components.

```
Eigenvalues of Centered Kernel Matrix
*************************************
```

```
 N  Eigenvalue  Difference  Proportion  Cumulative
 1  5.72097004  1.19980556  0.02888783  0.02888783
 2  4.52116448  0.23974587  0.02282945  0.05171728
 3  4.28141861  0.26090560  0.02161886  0.07333614
 4  4.02051301  0.56434211  0.02030143  0.09363758
 5  3.45617090  0.04657349  0.01745181  0.11108938
 6  3.40959740  0.24042161  0.01721664  0.12830602
```

```
Kernel PCA Number of Factors . . . . . . . . . . . . . . . . . 6
Variance Explained . . . . . . . . . . . . . . . . . . . 0.128306
Remaining Variance . . . . . . . . . . . . . . . . . . . 0.871694
Kernel Function. . . . . . . . . . . . . . . . . . . . . . . RBF2
Kernel Parameter [1] . . . . . . . . . . . . . . . . . . . . . 10
Total Number of Kernel Calls . . . . . . . . . . . . . . . . 20908
Total Processing Time. . . . . . . . . . . . . . . . . . . . . . 1
```

2. NIR Spectra data set: train: nr=21:
   First, read in the data:

```
  options NOECHO;
#include "..\\tdata\\nir.dat"
  options ECHO;
  nrtrn = nrow(xtrn); nctrn = ncol(xtrn);
  print "nrtrn,nctrn=",nrtrn,nctrn;
  xytrn = xtrn -> ytrn‘; /* attrib(xytrn); */
```

   Linear kernel:

```
    optn = [ "kern"    "line" ,
             "fact"         3 ,
             "pall"           ];
    < gof,pc,eval > = nlkpca(xtrn,optn);
```

```
                    *****************
                    Model Information
                    *****************

                Number Valid Observations   21
                Number Response Variables    0
                N Independend Variables    268
                Kernel Function         Linear
                Use Unscaled Data

          Memory needed for Kernel matrix: 0.00176239 Mb
```

```
          Eigenvalues of Centered Kernel Matrix
          *************************************

       N  Eigenvalue  Difference  Proportion  Cumulative
       1  131.200580  13.4245676  0.52053381  0.52053381
       2  117.776013  115.931634  0.46727229  0.98780610
       3  1.84437903  1.28061141  0.00731751  0.99512361
```

We only show the first five from the 21 columns:

```
          Eigenvectors of Centered Kernel Matrix
          **************************************

                  Dense Matrix (3 by 21)

    |          1          2          3          4          5
  ------------------------------------------------------------
PC1 |  -0.1450285 -0.0029272 -0.1659867  0.1284992 -0.0327733
PC2 |   0.4004388  0.3123114  0.3172391  0.2075480  0.2023632
PC3 |   0.5338241  0.2411152  0.0428607  0.0828897 -0.0584783

          Principal Components (Kernel  Linear)
          *************************************

                  Dense Matrix (3 by 21)

    |          1          2          3          4          5
  ------------------------------------------------------------
PC1 |  -0.1450285 -0.0029272 -0.1659867  0.1284992 -0.0327733
PC2 |   0.4004388  0.3123114  0.3172391  0.2075480  0.2023632
PC3 |   0.5338241  0.2411152  0.0428607  0.0828897 -0.0584783


Kernel PCA Number of Factors . . . . . . . . . . . . . . . . 3
Variance Explained . . . . . . . . . . . . . . . . . . 0.995124
Remaining Variance . . . . . . . . . . . . . . . . . . 0.00487639
Kernel Function. . . . . . . . . . . . . . . . . . . . . Linear
Total Number of Kernel Calls . . . . . . . . . . . . . . . 274
Total Processing Time. . . . . . . . . . . . . . . . . . . . 0
```

We illustrate the eigenvalue distribution of the RBF kernel matrix for only two values of `kfp1`:

```
optn = [ "kern"    "rbf2" ,
         "kfp1"       10. ,
```

```
         "fact"         3 ,
         "pall"           ];
< gof,pc,eval > = nlkpca(xtrn,optn);


           Eigenvalues of Centered Kernel Matrix
           *************************************

        N  Eigenvalue  Difference  Proportion  Cumulative
        1  1.00000001  6.661e-016  0.05000000  0.05000000
        2  1.00000001  2.220e-016  0.05000000  0.10000000
        3  1.00000001  2.220e-016  0.05000000  0.15000000
```

We only show the first five from the 21 columns:

```
           Eigenvectors of Centered Kernel Matrix
           **************************************

                   Dense Matrix (3 by 21)

    |          1          2          3          4          5
  ------------------------------------------------------------
PC1 |  0.1385307  0.1385307  0.1504903 -0.0370896 -0.0368392
PC2 | -0.1829113 -0.1829113  0.1227363  0.0958478  0.1066739
PC3 |  0.1326678  0.1326678 -0.2367299 -0.2686845 -0.2649538


            Principal Components (Kernel  RBF2)
            ***********************************

                   Dense Matrix (3 by 21)

    |          1          2          3          4          5
  ------------------------------------------------------------
PC1 |  0.1385307  0.1385307  0.1504903 -0.0370896 -0.0368392
PC2 | -0.1829113 -0.1829113  0.1227363  0.0958478  0.1066739
PC3 |  0.1326678  0.1326678 -0.2367299 -0.2686845 -0.2649538


Kernel PCA Number of Factors . . . . . . . . . . . . . . . . . 3
Variance Explained . . . . . . . . . . . . . . . . . . . . . 0.15
Remaining Variance . . . . . . . . . . . . . . . . . . . . . 0.85
Kernel Function. . . . . . . . . . . . . . . . . . . . . . . RBF2
Kernel Parameter [1] . . . . . . . . . . . . . . . . . . . . . 10
Total Number of Kernel Calls . . . . . . . . . . . . . . . . 274
Total Processing Time. . . . . . . . . . . . . . . . . . . . . 0
```

```
optn = [ "kern"    "rbf2" ,
         "kfp1"     .0001 ,
         "fact"         3 ,
         "pall"           ];
< gof,pc,eval > = nlkpca(xtrn,optn);


           Eigenvalues of Centered Kernel Matrix
           *************************************

        N  Eigenvalue  Difference  Proportion  Cumulative
        1  0.17068201  0.08428511  0.54004649  0.54004649
        2  0.08639690  0.03368047  0.27336415  0.81341064
        3  0.05271643  0.04893955  0.16679743  0.98020807


Kernel PCA Number of Factors . . . . . . . . . . . . . . . . . 3
Variance Explained . . . . . . . . . . . . . . . . . . . 0.980208
Remaining Variance . . . . . . . . . . . . . . . . . . . 0.0197919
Kernel Function. . . . . . . . . . . . . . . . . . . . . . . RBF2
Kernel Parameter [1] . . . . . . . . . . . . . . . . . . . 0.0001
Total Number of Kernel Calls . . . . . . . . . . . . . . . . 274
Total Processing Time. . . . . . . . . . . . . . . . . . . . . 0
```

3.5 Function `nlkpls`

< alpha,sres,vres,yptrn,yptnt > = nlkpls(trn,model,optn<,class<,tun<,kfun<,test>>>>)

Purpose: The `nlkpls` function implements a method for nonlinear Hilbert kernel partial least squares (PLS). In many practical aplications the RBF (radial base function kernel) has better modeling properties than the common linear function. The `nlkpls` function permits multiple runs for tuning kernel parameters using cross validation methods. It also permits the specification of a binary or multinomial response (target) using the `class` input argument. For multinomial response, the *one vs. all other* strategy is used for estimation and scoring.

Input: **trn** the $N \times M$ input data set containing the $N \times m$ predictor matrix $\mathbf{X}$ and the N response vector $\mathbf{Y}$. Which data columns are selected for $\mathbf{X}$ and $\mathbf{Y}$ must be specified with the model string.

model : The analysis model is specified in form of a string, e.g. `model= "3=1 2"`, containing column numbers for variables. The syntax of the `model` string argument is the same as for the `glmod()` function except for the additional *events / trial* response specification. **????**

optn : The option argument is specified in form of a two column matrix where the first column defines the option as string value (in quotes) and the second column can be used for a numeric or string specification of the option. Some of the options are similar to those of the `svm()` and `nlkpca()` functions. See table below for content.

class : This optional argument should be an integer scalar or vector of integer scalars naming the number of columns which are considered categorical (nominal scaled) variables.

tun This optional argument specifies a vector or matrix of grid points or ranges for the tuning of kernel function hyper parameters.

kfun This optional argument specifies the module name for a user specified kernel function. Note, the kernel matrix should satisfy the Mercer property of positive definiteness.

test specifies a matrix of test data which is not used for modeling but for predicted values are computed (scored) that can be returned with the last output argument `yptt`.

Options Matrix Argument: The option argument is specified in form of a two column matrix:

Option Name	Second Column	Meaning
"best"	int	only valid with Loo estimation: lists the *best* observations which when deleted improve most the goodness-of-fit
"block"	int	fold number for block cross validation
"cloo"		compute Loo (leave-one-out) estimates
"fact"	int	number p of factors (components)
"kern"		link function, default is linear
	"line"	linear function
	"poly"	polynomial function
	"rbf"	Gaussian radial basis function (def. scaling $[0, 1]$)
	"rbf2"	mod. Gaussian radial basis function (def. scaling $[0, 1]$)
	"erbf"	Exponential radial basis function
	"tanh"	sigmoid function (same as "sigm")
	"sigm"	sigmoid function (same as "tanh")
	"four"	Fourier function (def. scaling $[-\pi/2, \pi/2]$)
	"spli"	spline function (def. scaling $[0, 1]$)
	"anov"	anova function
	"curv"	curvspline function (def. scaling $[0, 1]$)
	"bspl"	Bspline function (def. scaling $[0, 1]$)
	"anob"	anova spline function (def. scaling $[0, 1]$)
	"ano1"	modified anova function (def. scaling $[0, 1]$)
	"ano2"	modified anova function (def. scaling $[0, 1]$)
	"ano3"	modified anova function (def. scaling $[0, 1]$)
"kfp1"	real	first parameter of kernel function
"kfp2"	real	second parameter of kernel function
"kfp3"	real	third parameter of kernel function
"meth"	"kpcr"	kernel principal components regression
	"kpls"	kernel partial least squares regression
	"dpls"	direct kernel partial least squares regression
"nopr"		no printed output in ...txt (but warnings or errors are in ...log)
"outl"	int	same as "best"
"pall"		is equivalent to "print" 10
"pmat"		print kernel matrix (this can be big)
"ppar"		print vector of parameter estimates α
"ppred"		print predicted values
"print"	int	amount of general printed output (default is 3)
"pshort"		less than default output
"psum"		much less than default output

"scale"	int	if not zero, perform scaling of data; this is default for several kernel functions
"split"	int	fold number for split cross validation
"tun"	"gsid"	perform grid search for tuning
	"patt"	perform pattern search for tuning
	"opti"	perform Melder-Mead optimization for tuning
"tunitr"	int	max iterations for tuning
"tuntim"	int	max time in seconds for tuning
"tunftol"	real	f termination for tuning
"tunxtol"	real	x termination for tuning
"tunsel"	"test"	for tuning use validation data evaluating goodness-of-fit (default if "tun" is specified)
	"train"	for tuning use validation data evaluating goodness-of-fit (default if "tun" is not specified)

Some of the kernel functions, especially those based on the `exp` function, like all `rbf` functions, can create kernel matrices which are diagonal or even identity matrices for extrem parameter settings. using the identity matrix for the kernel matrix produces useless results.

Output: alpha vector or matrix with parameter estimates

sres a vector of scalar values

vres for multinomial response the `sres` contains in rows information for each response level vs. all other

yptrn vector of predicted values for training data

yptst vector os predicted values of test data

Restrictions: 1. The input data set should not contain missing values, string or complex data.

Relationships: nlkpca(), pls(), svm()

Examples: 1. Australian Institute of Sport (ais) data set (Weisberg, 2002):

```
options NOECHO;
ais = [
%inc "..\\tdata\\ais.dat";
];
options ECHO;
cname = [ "Sex" "Ht" "Wt" "LBM" "RCC" "WCC" "Hc" "Hg" ];
ais = shape(ais,.,8);
nobs = nrow(ais); nvar = ncol(ais);
ais = ais -> log(ais[,5]);
```

Using the "nlkpls" method with linear kernel:

```
modl = "4 = 2 3 9 6";
optn = [ "slic"         8 ,
         "dec"      "svd" ,
         "numdir"       4 ,
         "print"        3 ,
         "nperm"      499 ,
         "seed"       123 ];
< gof,parm,atst,ptst > = nlkpls(ais,modl,optn);
```

Fact	SSX	Xdif	Ydif
0	2789392633	.	.
1	30949465.7	2758443167	192.058321
2	426646.390	30522819.3	0.03416301
3	2.72599730	426643.664	0.01018565

```
 Kernel PLS Number of Factors . . . . . . . . . . . . . . . . . 4
 Mean Squared Error (MSE, Training Data). . . . . . . . . 13.9221
 Mean Absolute Error (MAE, Training Data) . . . . . . .  2.9882
 Average Positive Loss (Training Data). . . . .  3.42964 (N=88)
 Average Negative Loss (Training Data). . . . . 2.64744 (N=114)
 Kernel Function. . . . . . . . . . . . . . . . . . . . . Linear
 Total Number of Kernel Calls . . . . . . . . . . . . . . 20908
 Total Processing Time. . . . . . . . . . . . . . . . . . . . 1
 Linear Kernel Constant (PCE) . . . . . . . . . . . . . . 64.8737
```

```
 y = ais[,yind]; nr = nrow(ais);
 res = y - yptr';
 s1 = ssq(res); s2 = s1 / nr;
 print "sse, mse=", s1, s2;
```

```
sse, mse= 2812.27 13.922
```

This is the traditional PLS call:

```
 optn = [ "prin"         1 ,
          "meth"     "pls" ,
          "tech"  "krnpls" ];
 yind = 4; xind = [ 2 3 9 6 ];
 cmp  = [ 1:4 ];
 < rms,bmat,yprd > = pls(ais,yind,xind,cmp,optn);
```

```
                Analysis of Training Data
                -------------------------

                 Summary Table: Training
                 -----------------------

   Factor    X_VarExp    Y_VarExp         RMS          R2
        1  0.89501676  0.87706842  4.57126303  0.87706842
        2  0.98884248  0.88123227  4.49317899  0.88123227
        3  0.99997186  0.88243891  4.47029602  0.88243891
        4  1.00000000  0.91809760  3.73123672  0.91809760
```

```
y = ais[,yind]; nr = nrow(ais);
res = y - yprd[,4];
s1 = ssq(res); s2 = s1 / nr;
print "sse, mse=", s1, s2;
```

As you can see we obtain the same result, however, much faster with the specific `pls` call than with the more general `nlkpls` call:

```
sse, mse= 2812.27 13.922
```

Now, use the RBF kernel:

```
optn = [ "meth"    "kpls" ,
         "kern"    "rbf2" ,
         "fact"         4 ,
         "ppred"          ,
         "pall"           ];
modl = "4 = 2 3 9 6";
tun = [ .01 .1 1. 10. ];
< alpha,sres,vres,yptr > = nlkpls(ais,modl,optn,.,tun);
```

```
             Best 4 Results of Grid Search
              ----------------------------

          N        KF1      L1err       Error    TrainErr
          4  10.000000 7.67859361  1551.07591  7.67859361
          2  0.1000000 12.7083058  2567.07776  12.7083058
          1  0.0100000 13.6464396  2756.58079  13.6464396
          3  1.0000000 13.9173195  2811.29854  13.9173195

           Fact         SSX        Xdif        Ydif
```

```
  0  763.949504      .           .
  1  627.177013  136.772490  185.013336
  2  577.933090  49.2439234  0.67091219
  3  522.161312  55.7717779  0.58694927

 Kernel PLS Number of Factors . . . . . . . . . . . . . . . . . 4
 Mean Squared Error (MSE, Training Data). . . . . . . . . 7.67859
 Mean Absolute Error (MAE, Training Data) . . . . . . . . 2.10641
 Average Positive Loss (Training Data). . . . .  2.17089 (N=98)
 Average Negative Loss (Training Data). . . . . 2.04565 (N=104)
 Kernel Function. . . . . . . . . . . . . . . . . . . . . . RBF2
 Kernel Parameter [1] . . . . . . . . . . . . . . . . . . . . 10
 Total Number of Kernel Calls . . . . . . . . . . . . . . 102921
 Total Processing Time. . . . . . . . . . . . . . . . . . . . 3
 Linear Kernel Constant (PCE) . . . . . . . . . . . . . . 64.8737


  y = ais[,yind]; nr = nrow(ais);
  res = y - yptr';
  s1 = ssq(res); s2 = s1 / nr;
  print "sse, mse=", s1, s2;


sse, mse= 1551.08 7.6786
```

2. Boston Housing: nobs=506, nvar=14:
   First, read in the data:

```
fid = fopen("..\\tdata\\housing.dat","r");
form = "%g %g %g %g %g %g %g %g %g %g %g %g %g %g";
hous = fscanf(fid,form,506,14);
vnam = [ "crim" "zn" "indus" "chas" "nox" "rm" "age"
         "dis" "rad" "tax" "ptrat" "b" "lstat" "medv" ];
hous = cname(hous,vnam);
nr = nrow(hous);
/* print hous[1:10,; */


  optn = [ "meth"    "kpls" ,
           "kern"    "line" ,
           "fact"        10 ,
           "ppred"          ,
           "pall"           ];
  modl = "14 = 1 : 13";
  < alpha,sres,vres,yptr > = nlkpls(hous,modl,optn);
```

```
 Fact         SSX        Xdif        Ydif
    0  2.535e+014           .           .
    1  1.023e+013  2.433e+014  466.876984
    2  7.942e+011  9.437e+012  0.03616631
    3  2.056e+010  7.736e+011  0.07118562
    4  8591407737  1.197e+010  0.30250690
    5   611679109  7979728627  0.19382942
    6   107150897   504528213  0.06141443
    7  64976679.5  42174217.1  0.11101507
    8  28871952.7  36104726.9  0.05344789
    9  918237.801  27953714.9  0.02281388
```

```
Kernel PLS Number of Factors . . . . . 10
Mean Squared Error (MSE, Training Data). . . . . 24.4258
Mean Absolute Error (MAE, Training Data) . . . . . 3.50328
Average Positive Loss (Training Data). . . . . 2.91556 (N=304)
Average Negative Loss (Training Data). . . . . 4.38777 (N=202)
Kernel Function. . . . . Linear
Total Number of Kernel Calls . . . . . 129284
Total Processing Time. . . . . 33
Linear Kernel Constant (PCE) . . . . . 22.5328
```

```
y = hous[,yind]; nr = nrow(hous);
res = y - yptr‘;
s1 = ssq(res); s2 = s1 / nr;
print "sse, mse=", s1, s2;
```

```
sse, mse= 12359.45 24.426
```

Compared to that we run the traditional PLS algorithm:

```
**************************
Kernel Partial Least Squares
**************************

X Dimension . . . . . 13
Y Dimension . . . . . 1
Nobs Training Data. . . . . 506
Number Analyses . . . . . 10
Maximum Number Factors. . . . . 10

-------------------------
Analysis of Training Data
```

```
-------------------------

 Summary Table: Training
 -----------------------

Factor    X_VarExp    Y_VarExp         RMS          R2
     1  0.80508818  0.24227210  7.99794054  0.24227210
     2  0.94450953  0.26942855  7.85331254  0.26942855
     3  0.98972248  0.32050921  7.57379108  0.32050921
     4  0.99344496  0.51051474  6.42822904  0.51051474
     5  0.99802224  0.60079391  5.80523907  0.60079391
     6  0.99913209  0.62493450  5.62697636  0.62493450
     7  0.99940679  0.66541682  5.31463678  0.66541682
     8  0.99967974  0.68306063  5.17260871  0.68306063
     9  0.99994045  0.69025001  5.11360509  0.69025001
    10  0.99998521  0.71066198  4.94224519  0.71066198


 Time for Training: 0
     Total Time: 0

sse, mse= 12359.45 24.426
```

Now we run `kpls` using RBF Kernel with different values of the bandwidth parameter:

```
optn = [ "meth"    "kpls" ,
         "kern"    "rbf2" ,
         "fact"        10 ,
         "ppred"          ,
         "pall"          ];
modl = "14 = 1 : 13";
tun = [ .01 .1 1. 10. ];
< alpha,sres,vres,yptr > = nlkpls(hous,modl,optn,.,tun);
```

```
 Best 4 Results of Grid Search
 -----------------------------

N        KF1      L1err       Error    TrainErr
4  10.000000 0.10003232  50.6163528  0.10003232
3  1.0000000 3.59081102  1816.95038  3.59081102
2  0.1000000 8.57670852  4339.81451  8.57670852
1  0.0100000 12.6576099  6404.75059  12.6576099
```

The best result is obtained for band width 10:

```
            Grid Search Results: L2 Error (Training)
             ****************************************

                    Dense Row Vector (ncol=4)

          R |          1          2          3          4
                6404.7506  4339.8145  1816.9504  50.616353

              Fact         SSX        Xdif        Ydif
                 0  915.229759           .           .
                 1  852.813326  62.4164332  477.240560
                 2  812.757823  40.0555034  0.84709880
                 3  793.254145  19.5036779  0.91879709
                 4  771.593443  21.6607018  0.75298552
                 5  752.876496  18.7169466  0.60005021
                 6  733.864648  19.0118480  0.35182905
                 7  721.197304  12.6673442  0.29731449
                 8  709.227252  11.9700523  0.32471183
                 9  702.624902  6.60234949  0.36547887

   Kernel PLS Number of Factors . . . . . 10
   Mean Squared Error (MSE, Training Data). . . . . 0.100032
   Mean Absolute Error (MAE, Training Data) . . . . . 0.188909
   Average Positive Loss (Training Data). . . . 0.187428 (N=255)
   Average Negative Loss (Training Data). . . . 0.190415 (N=251)
   Kernel Function. . . . . RBF2
   Kernel Parameter [1] . . . . . 10
   Total Number of Kernel Calls . . . . . 642369
   Total Processing Time. . . . . 153
   Linear Kernel Constant (PCE) . . . . . 22.5328


    y = hous[,yind];
    res = y - yptr`; nr = nrow(hous);
    s1 = ssq(res); s2 = s1 / nr;
    print "sse, mse=", s1, s2;


sse, mse= 50.616 0.1000
```