

## Function sbr

### Description

Function for scalable Bayesian regression (SBR/SSBR) in normal linear models with multiple types (sources) of feature matrices (with  $K$  being the number of sources). For details see Perrakis and Mukherjee (2017).

### Usage

```
sbr(y, X, trX, G, estimator = 'PM', sparsify = FALSE, sparse.control = 1, p.threshold = 5000,
cov.blocks = 1000, parallel = FALSE, cl, L.optim = 10-4, U.optim = 104)
```

### Arguments

|                             |   |
|-----------------------------|---|
| <code>y</code>              | a standardized response vector.   |
| <code>X</code>              | a standardized feature matrix (if $K = 1$ ) or a list of standardized feature matrices (if $K > 1$ ).   |
| <code>trX</code>            | (optional) the transpose matrix of <code>X</code> (if $K = 1$ ) or a list of transpose matrices (if $K > 1$ ).  |
| <code>G</code>              | the inner-product Gram matrix (if $K = 1$ ) or a list containing the multiple Gram matrices (if $K > 1$ ).  |
| <code>estimator</code>      | the estimator used for tuning the shrinkage levels. Available estimates are leave-one-out cross-validation ("CV"), maximum marginal likelihood ("ML") and the posterior mode ("PM", default).   |
| <code>sparsify</code>       | logical, if <code>TRUE</code> the SSBR solution is calculated, default option is <code>FALSE</code> .   |
| <code>sparse.control</code> | numerical value for controlling the effect of sample size ( $n$ ) on the resulting SSBR solution. Default option is 1 (no control). A recommended option for sparser solutions is <code>sparse.control = log(n)</code> .  |
| <code>p.threshold</code>    | used when <code>sparsify = TRUE</code> . When the number of source-specific features exceeds <code>p.threshold</code> , <code>sbr</code> will automatically use block matrix computations (see the following argument). Default option is 5000.                                       |
| <code>cov.blocks</code>     | used for block-matrix computation of the main diagonal of the covariance matrix when <code>sparsify = TRUE</code> . Argument <code>cov.blocks</code> corresponds to block size (not the number of blocks). Default option is 1000, i.e. blocks of dimensionality $1000 \times 1000$ . |
| <code>parallel</code>       | logical, if <code>parallel = TRUE</code> the calculation of variance components required for the SSBR solution is performed in parallel. Default is <code>FALSE</code> .  |
| <code>cl</code>             | the number of cores to use when <code>parallel = TRUE</code> . Must be provided by the user.  |

|         |  |
|---------|--|
| L.optim | lower bound for the optimization procedure used to tune the shrinkage levels, default is $10^{-4}$ . |
| U.optim | upper bound for the optimization procedure used to tune the shrinkage levels, default is $10^4$ .    |

## Value

|                 |  |
|-----------------|--|
| BetaSBR         | the SBR regression coefficients (at the posterior mode).               |
| BetaSSBR        | the SSBR regression coefficients (when <code>sparsify = TRUE</code> ). |
| Sigma2          | the variance component (at the posterior mode).                        |
| Lambda          | the vector of penalty parameters.                                      |
| LambdaEstimator | the estimator used for <code>Lambda</code> .                           |
| Duration        | runtime.   |

## References

Perrakis, K. and Mukherjee, S. (2017). *Scalable Bayesian regression in high dimensions with multiple data sources*, arXiv:1710.00596v2 [stat.ME].

## Function gram

### Description

Function for calculating the (inner-product) Gram matrix that allows for block-matrix multiplication.

### Usage

```
gram(X, trX, block = FALSE, block.size = 1000)
```

### Arguments

|            |  |
|------------|--|
| X          | a standardized feature matrix.   |
| trX        | (optional) the transpose matrix of X.  |
| block      | logical, block matrix computation is performed when TRUE, default option is FALSE.                               |
| block.size | used when <code>block = TRUE</code> . Default option is 1000, i.e. blocks of dimensionality $1000 \times 1000$ . |

## Value

Returns the inner-product Gram matrix.

## Function `gram.parallel`

### Description

Function for calculating the (inner-product) Gram matrix that allows for block-matrix multiplication performed in parallel.

### Usage

```
gram.parallel(X, cl, ...)
```

### Arguments

`X` a standardized feature matrix.

`cl` the number of cores to use. Must be provided by the user.

`...` additional arguments passed from function `gram`.

### Value

Returns the inner-product Gram matrix.