stepwiseglm

Create generalized linear regression model by stepwise regression

Syntax

```
mdl = stepwiseglm(tbl)
mdl = stepwiseglm(X,y)
mdl = stepwiseglm(___,modelspec)
mdl = stepwiseglm(___,modelspec,Name,Value)
```

Description

`mdl = stepwiseglm(tbl)` creates a generalized linear model of a table or dataset array `tbl` using stepwise regression to add or remove predictors, starting from a constant model. `stepwiseglm` uses the last variable of `tbl` as the response variable. `stepwiseglm` uses forward and backward stepwise regression to determine a final model. At each step, the function searches for terms to add the model to or remove from the model, based on the value of the `'Criterion'` argument.

`mdl = stepwiseglm(X,y)` creates a generalized linear model of the responses `y` to a data matrix `X`. example

`mdl = stepwiseglm(___,modelspec)` specifies the starting model `modelspec` using any of the input argument combinations in previous syntaxes.

`mdl = stepwiseglm(___,modelspec,Name,Value)` specifies additional options using one or more name-value pair arguments. For example, you can specify the categorical variables, the smallest or largest set of terms to use in the model, the maximum number of steps to take, or the criterion that `stepwiseglm` uses to add or remove terms. example

Examples

collapse all

Generalized Linear Model Using Stepwise Algorithm

Create response data using just three of 20 predictors, and create a generalized linear model using stepwise algorithm to see if it uses just the correct predictors.

View MATLAB Command

Create data with 20 predictors, and Poisson response using just three of the predictors, plus a constant.

```
rng('default') % for reproducibility
X = randn(100,20);
mu = exp(X(:,[5 10 15])*[.4;.2;.3] + 1);
y = poissrnd(mu);
```

Fit a generalized linear model using the Poisson distribution.

```
mdl =  stepwiseglm(X,y,...
    'constant','upper','linear','Distribution','poisson')
```

```
1. Adding x5, Deviance = 134.439, Chi2Stat = 52.24814, PValue = 4.891229e-13
2. Adding x15, Deviance = 106.285, Chi2Stat = 28.15393, PValue = 1.1204e-07
```

```
3. Adding x10, Deviance = 95.0207, Chi2Stat = 11.2644, PValue = 0.000790094
mdl = 
Generalized linear regression model:
    log(y) ~ 1 + x5 + x10 + x15
    Distribution = Poisson

Estimated Coefficients:
                   Estimate       SE        tStat       pValue  
                   ________    ________    ______    ___________

    (Intercept)     1.0115     0.064275    15.737     8.4217e-56
    x5             0.39508     0.066665    5.9263     3.0977e-09
    x10            0.18863      0.05534    3.4085      0.0006532
    x15            0.29295     0.053269    5.4995     3.8089e-08


100 observations, 96 error degrees of freedom
Dispersion: 1
Chi^2-statistic vs. constant model: 91.7, p-value = 9.61e-20
```

The starting model is the constant model. `stepwiseglm` by default uses deviance of the model as the criterion. It first adds `x5` into the model, as the p-value for the test statistic, deviance (the differences in the deviances of the two models), is less than the default threshold value 0.05. Then, it adds `x15` because given `x5` is in the model, when `x15` is added, the p-value for chi-squared test is smaller than 0.05. It then adds `x10` because given `x5` and `x15` are in the model, when `x10` is added, the p-value for the chi-square test statistic is again less than 0.05.

Input Arguments

collapse all

`tbl` — Input data

table | dataset array

Input data including predictor and response variables, specified as a table or dataset array. The predictor variables and response variable can be numeric, logical, categorical, character, or string. The response variable can have a data type other than numeric only if `'Distribution'` is `'binomial'`.

- By default, `stepwiseglm` takes the last variable as the response variable and the others as the predictor variables.
- To set a different column as the response variable, use the `ResponseVar` name-value pair argument.
- To use a subset of the columns as predictors, use the `PredictorVars` name-value pair argument.
- To define a model specification, set the `modelspec` argument using a formula or terms matrix. The formula or terms matrix specifies which columns to use as the predictor or response variables.

The variable names in a table do not have to be valid MATLAB® identifiers. However, if the names are not valid, you cannot use a formula when you fit or adjust a model; for example:

- You cannot specify `modelspec` using a formula.
- You cannot use a formula to specify the terms to add or remove when you use the `addTerms` function or the `removeTerms` function, respectively.

- You cannot use a formula to specify the lower and upper bounds of the model when you use the `step` or `stepwiseglm` function with the name-value pair arguments `'Lower'` and `'Upper'`, respectively.

You can verify the variable names in `tbl` by using the `isvarname` function. If the variable names are not valid, then you can convert them by using the `matlab.lang.makeValidName` function.

`X` — Predictor variables

matrix

Predictor variables, specified as an n-by-p matrix, where n is the number of observations and p is the number of predictor variables. Each column of `X` represents one variable, and each row represents one observation.

By default, there is a constant term in the model, unless you explicitly remove it, so do not include a column of 1s in `X`.

Data Types: `single` | `double`

`y` — Response variable

vector | matrix

Response variable, specified as a vector or matrix.

- If `'Distribution'` is not `'binomial'`, then `y` must be an n-by-1 vector, where n is the number of observations. Each entry in `y` is the response for the corresponding row of `X`. The data type must be single or double.
- If `'Distribution'` is `'binomial'`, then `y` can be an n-by-1 vector or n-by-2 matrix with counts in column 1 and `BinomialSize` in column 2.

Data Types: `single` | `double` | `logical` | `categorical`

`modelspec` — Starting model

`'constant'` (default) | character vector or string scalar naming the model | t-by-(p + 1) terms matrix | character vector or string scalar formula in the form `'y ~ terms'`

Starting model for `stepwiseglm`, specified as one of the following:

- A character vector or string scalar naming the model.

Value	**Model Type**
`'constant'`	Model contains only a constant (intercept) term.
`'linear'`	Model contains an intercept and linear term for each predictor.
`'interactions'`	Model contains an intercept, linear term for each predictor, and all products of pairs of distinct predictors (no squared terms).

Value	Model Type
`'purequadratic'`	Model contains an intercept term and linear and squared terms for each predictor.
`'quadratic'`	Model contains an intercept term, linear and squared terms for each predictor, and all products of pairs of distinct predictors.
`'polyijk'`	Model is a polynomial with all terms up to degree *i* in the first predictor, degree *j* in the second predictor, and so on. Specify the maximum degree for each predictor by using numerals 0 though 9. The model contains interaction terms, but the degree of each interaction term does not exceed the maximum value of the specified degrees. For example, `'poly13'` has an intercept and x_1, x_2, x_2^2, x_2^3, x_1*x_2, and $x_1*x_2^2$ terms, where x_1 and x_2 are the first and second predictors, respectively.

- A t-by-(p + 1) matrix, or a Terms Matrix, specifying terms in the model, where t is the number of terms and p is the number of predictor variables, and +1 accounts for the response variable. A terms matrix is convenient when the number of predictors is large and you want to generate the terms programmatically.
- A character vector or string scalar Formula in the form

  ```
  'y ~ terms',
  ```

  where the `terms` are in Wilkinson Notation. The variable names in the formula must be variable names in `tbl` or variable names specified by `Varnames`. Also, the variable names must be valid MATLAB identifiers.

  The software determines the order of terms in a fitted model by using the order of terms in `tbl` or `X`. Therefore, the order of terms in the model can be different from the order of terms in the specified formula.

If you want to specify the smallest or largest set of terms in the model that `stepwiselm` fits, use the `Lower` and `Upper` name-value pair arguments.

Data Types: `char` | `string` | `single` | `double`

Name-Value Pair Arguments

Specify optional comma-separated pairs of `Name,Value` arguments. `Name` is the argument name and `Value` is the corresponding value. `Name` must appear inside quotes. You can specify several name and value pair arguments in any order as `Name1,Value1,...,NameN,ValueN`.

Example: `'Criterion','aic','Distribution','poisson','Upper','interactions'` specifies Akaike Information Criterion as the criterion to add or remove variables to the model, Poisson distribution as the distribution of the response variable, and a model with all possible interactions as the largest model to consider as the fit.

`'BinomialSize'` — Number of trials for binomial distribution

1 (default) | numeric scalar | numeric vector | character vector | string scalar

Number of trials for binomial distribution, that is the sample size, specified as the comma-separated pair consisting of `'BinomialSize'` and the variable name in `tbl`, a numeric scalar, or a numeric vector of the same length as the response. This is the parameter `n` for the fitted binomial distribution. `BinomialSize` applies only when the `Distribution` parameter is `'binomial'`.

If `BinomialSize` is a scalar value, that means all observations have the same number of trials.

As an alternative to `BinomialSize`, you can specify the response as a two-column matrix with counts in column 1 and `BinomialSize` in column 2.

Data Types: `single` | `double` | `char` | `string`

`'CategoricalVars'` — Categorical variable list

string array | cell array of character vectors | logical or numeric index vector

Categorical variable list, specified as the comma-separated pair consisting of `'CategoricalVars'` and either a string array or cell array of character vectors containing categorical variable names in the table or dataset array `tbl`, or a logical or numeric index vector indicating which columns are categorical.

- If data is in a table or dataset array `tbl`, then, by default, `stepwiseglm` treats all categorical values, logical values, character arrays, string arrays, and cell arrays of character vectors as categorical variables.
- If data is in matrix `X`, then the default value of `'CategoricalVars'` is an empty matrix `[]`. That is, no variable is categorical unless you specify it as categorical.

For example, you can specify the second and third variables out of six as categorical using either of the following:

Example: `'CategoricalVars',[2,3]`

Example: `'CategoricalVars',logical([0 1 1 0 0 0])`

Data Types: `single` | `double` | `logical` | `string` | `cell`

`'Criterion'` — Criterion to add or remove terms

`'Deviance'` (default) | `'sse'` | `'aic'` | `'bic'` | `'rsquared'` | `'adjrsquared'`

Criterion to add or remove terms, specified as the comma-separated pair consisting of `'Criterion'` and one of these values:

- `'Deviance'` — p-value for an F-test or chi-squared test of the change in the deviance that results from adding or removing the term. The F-test tests a single model, and the chi-squared test compares two different models.
- `'sse'` — p-value for an F-test of the change in the sum of squared error that results from adding or removing the term.
- `'aic'` — Change in the value of the Akaike information criterion (AIC).
- `'bic'` — Change in the value of the Bayesian information criterion (BIC).
- `'rsquared'` — Increase in the value of R^2.
- `'adjrsquared'` — Increase in the value of adjusted R^2.

Example: `'Criterion','bic'`

`'DispersionFlag'` — Indicator to compute dispersion parameter

`false` for `'binomial'` and `'poisson'` distributions (default) | `true`

Indicator to compute dispersion parameter for `'binomial'` and `'poisson'` distributions, specified as the comma-separated pair consisting of `'DispersionFlag'` and one of the following.

`true`	Estimate a dispersion parameter when computing standard errors. The estimated dispersion parameter value is the sum of squared Pearson residuals divided by the degrees of freedom for error (DFE).
`false`	Default. Use the theoretical value of 1 when computing standard errors.

The fitting function always estimates the dispersion for other distributions.

Example: `'DispersionFlag',true`

`'Distribution'` — Distribution of the response variable

`'normal'` (default) | `'binomial'` | `'poisson'` | `'gamma'` | `'inverse gaussian'`

Distribution of the response variable, specified as the comma-separated pair consisting of `'Distribution'` and one of the following.

`'normal'`	Normal distribution
`'binomial'`	Binomial distribution
`'poisson'`	Poisson distribution
`'gamma'`	Gamma distribution
`'inverse gaussian'`	Inverse Gaussian distribution

Example: `'Distribution','gamma'`

`'Exclude'` — Observations to exclude

logical or numeric index vector

Observations to exclude from the fit, specified as the comma-separated pair consisting of `'Exclude'` and a logical or numeric index vector indicating which observations to exclude from the fit.

For example, you can exclude observations 2 and 3 out of 6 using either of the following examples.

Example: `'Exclude',[2,3]`

Example: `'Exclude',logical([0 1 1 0 0 0])`

Data Types: `single` | `double` | `logical`

`'Intercept'` — Indicator for constant term

`true` (default) | `false`

Indicator for the constant term (intercept) in the fit, specified as the comma-separated pair consisting of `'Intercept'` and either `true` to include or `false` to remove the constant term from the model.

Use `'Intercept'` only when specifying the model using a character vector or string scalar, not a formula or matrix.

Example: `'Intercept',false`

`'Link'` — Link function

canonical link function (default) | scalar value | structure

Link function to use in place of the canonical link function, specified as the comma-separated pair consisting of `'Link'` and one of the following.

Link Function Name	Link Function	Mean (Inverse) Function
`'identity'`	$f(\mu) = \mu$	$\mu = Xb$
`'log'`	$f(\mu) = \log(\mu)$	$\mu = \exp(Xb)$
`'logit'`	$f(\mu) = \log(\mu/(1–\mu))$	$\mu = \exp(Xb) / (1 + \exp(Xb))$
`'probit'`	$f(\mu) = \Phi^{-1}(\mu)$, where Φ is the cumulative distribution function of the standard normal distribution.	$\mu = \Phi(Xb)$
`'comploglog'`	$f(\mu) = \log(–\log(1 – \mu))$	$\mu = 1 – \exp(–\exp(Xb))$
`'reciprocal'`	$f(\mu) = 1/\mu$	$\mu = 1/(Xb)$
`p` (a number)	$f(\mu) = \mu^p$	$\mu = Xb^{1/p}$
`S` (a structure) with three fields. Each field holds a function handle that accepts a vector of inputs and returns a vector of the same size: • `S.Link` — The link function • `S.Inverse` — The inverse link function • `S.Derivative` — The derivative of the link function	$f(\mu)$ = `S.Link`(μ)	μ = `S.Inverse`(Xb)

The link function defines the relationship $f(\mu) = X^*b$ between the mean response μ and the linear combination of predictors X^*b.

For more information on the canonical link functions, see Canonical Function.

Example: `'Link','probit'`

Data Types: `char` | `string` | `single` | `double` | `struct`

`'Lower'` — Model specification describing terms that cannot be removed from model

`'constant'` (default) | character vector | string scalar | terms matrix

Model specification describing terms that cannot be removed from the model, specified as the comma-separated pair consisting of `'Lower'` and one of the options for `modelspec` naming the model.

Example: `'Lower','linear'`

`'NSteps'` — Maximum number of steps to take

no limit (default) | positive integer

Maximum number of steps to take, specified as the comma-separated pair consisting of `'NSteps'` and a positive integer.

Example: `'NSteps',5`

Data Types: `single` | `double`

`'Offset'` — Offset variable

[ ] (default) | numeric vector | character vector | string scalar

Offset variable in the fit, specified as the comma-separated pair consisting of `'Offset'` and the variable name in `tbl` or a numeric vector with the same length as the response.

`stepwiseglm` uses `Offset` as an additional predictor with a coefficient value fixed at 1. In other words, the formula for fitting is

$f(\mu)$ = `Offset` + $X*b$,

where f is the link function, μ is the mean response, and $X*b$ is the linear combination of predictors X. The `Offset` predictor has coefficient `1`.

For example, consider a Poisson regression model. Suppose the number of counts is known for theoretical reasons to be proportional to a predictor `A`. By using the log link function and by specifying `log(A)` as an offset, you can force the model to satisfy this theoretical constraint.

Data Types: `single` | `double` | `char` | `string`

`'PEnter'` — Threshold for criterion to add term

scalar value

Threshold for the criterion to add a term, specified as the comma-separated pair consisting of `'PEnter'` and a scalar value, as described in this table.

Criterion	Default Value	Decision
`'Deviance'`	0.05	If the p-value of the F-statistic or chi-squared statistic is less than `PEnter` (p-value to enter), add the term to the model.
`'SSE'`	0.05	If the p-value of the F-statistic is less than `PEnter`, add the term to the model.
`'AIC'`	0	If the change in the AIC of the model is less than `PEnter`, add the term to the model.
`'BIC'`	0	If the change in the BIC of the model is less than `PEnter`, add the term to the model.
`'Rsquared'`	0.1	If the increase in the R-squared value of the model is greater than `PEnter`, add the term to the model.
`'AdjRsquared'`	0	If the increase in the adjusted R-squared value of the model is greater than `PEnter`, add the term to the model.

For more information, see the `Criterion` name-value pair argument.

Example: `'PEnter',0.075`

`'PredictorVars'` — Predictor variables

string array | cell array of character vectors | logical or numeric index vector

Predictor variables to use in the fit, specified as the comma-separated pair consisting of `'PredictorVars'` and either a string array or cell array of character vectors of the variable names in the table or dataset array `tbl`, or a logical or numeric index vector indicating which columns are predictor variables.

The string values or character vectors should be among the names in `tbl`, or the names you specify using the `'VarNames'` name-value pair argument.

The default is all variables in `X`, or all variables in `tbl` except for `ResponseVar`.

For example, you can specify the second and third variables as the predictor variables using either of the following examples.

Example: `'PredictorVars',[2,3]`

Example: `'PredictorVars',logical([0 1 1 0 0 0])`

Data Types: `single` | `double` | `logical` | `string` | `cell`

`'PRemove'` — Threshold for criterion to remove term

scalar value

Threshold for the criterion to remove a term, specified as the comma-separated pair consisting of `'PRemove'` and a scalar value, as described in this table.

Criterion	Default Value	Decision
`'Deviance'`	0.10	If the p-value of the F-statistic or chi-squared statistic is greater than `PRemove` (p-value to remove), remove the term from the model.
`'SSE'`	0.10	If the p-value of the F-statistic is greater than `PRemove`, remove the term from the model.
`'AIC'`	0.01	If the change in the AIC of the model is greater than `PRemove`, remove the term from the model.
`'BIC'`	0.01	If the change in the BIC of the model is greater than `PRemove`, remove the term from the model.
`'Rsquared'`	0.05	If the increase in the R-squared value of the model is less than `PRemove`, remove the term from the model.
`'AdjRsquared'`	-0.05	If the increase in the adjusted R-squared value of the model is less than `PRemove`, remove the term from the model.

At each step, the `stepwiseglm` function also checks whether a term is redundant (linearly dependent) with other terms in the current model. When a term is linearly dependent on other terms in the current model, the `stepwiseglm` function removes the redundant term, regardless of the criterion value.

For more information, see the `Criterion` name-value pair argument.

Example: `'PRemove',0.05`

`'ResponseVar'` — Response variable

last column in `tbl` (default) | character vector or string scalar containing variable name | logical or numeric index vector

Response variable to use in the fit, specified as the comma-separated pair consisting of `'ResponseVar'` and either a character vector or string scalar containing the variable name in the table or dataset array `tbl`, or a logical or numeric index vector indicating which column is the response variable. You typically need to use `'ResponseVar'` when fitting a table or dataset array `tbl`.

For example, you can specify the fourth variable, say `yield`, as the response out of six variables, in one of the following ways.

Example: `'ResponseVar','yield'`

Example: `'ResponseVar',[4]`

Example: `'ResponseVar',logical([0 0 0 1 0 0])`

Data Types: `single` | `double` | `logical` | `char` | `string`

`'Upper'` — Model specification describing largest set of terms in fit

`'interactions'` (default) | character vector | string scalar | terms matrix

Model specification describing the largest set of terms in the fit, specified as the comma-separated pair consisting of `'Upper'` and one of the options for `modelspec` naming the model.

Example: `'Upper','quadratic'`

`'VarNames'` — Names of variables

`{'x1','x2',...,'xn','y'}` (default) | string array | cell array of character vectors

Names of variables, specified as the comma-separated pair consisting of `'VarNames'` and a string array or cell array of character vectors including the names for the columns of `X` first, and the name for the response variable `y` last.

`'VarNames'` is not applicable to variables in a table or dataset array, because those variables already have names.

The variable names do not have to be valid MATLAB identifiers. However, if the names are not valid, you cannot use a formula when you fit or adjust a model; for example:

- You cannot use a formula to specify the terms to add or remove when you use the `addTerms` function or the `removeTerms` function, respectively.
- You cannot use a formula to specify the lower and upper bounds of the model when you use the `step` or `stepwiseglm` function with the name-value pair arguments `'Lower'` and `'Upper'`, respectively.

Before specifying `'VarNames',varNames`, you can verify the variable names in `varNames` by using the `isvarname` function. If the variable names are not valid, then you can convert them by using the `matlab.lang.makeValidName` function.

Example: `'VarNames',{'Horsepower','Acceleration','Model_Year','MPG'}`

Data Types: `string` | `cell`

'Verbose' — Control for display of information
1 (default) | 0 | 2

Control for the display of information, specified as the comma-separated pair consisting of 'Verbose' and one of these values:

- 0 — Suppress all display.
- 1 — Display the action taken at each step.
- 2 — Display the evaluation process and the action taken at each step.

Example: 'Verbose',2

'Weights' — Observation weights
ones(n,1) (default) | n-by-1 vector of nonnegative scalar values

Observation weights, specified as the comma-separated pair consisting of 'Weights' and an n-by-1 vector of nonnegative scalar values, where n is the number of observations.

Data Types: single | double

Output Arguments

collapse all

mdl — Generalized linear regression model
GeneralizedLinearModel object

Generalized linear regression model, specified as a GeneralizedLinearModel object created using fitglm or stepwiseglm.

More About

collapse all

Terms Matrix

A terms matrix T is a t-by-(p + 1) matrix specifying terms in a model, where t is the number of terms, p is the number of predictor variables, and +1 accounts for the response variable. The value of T(i,j) is the exponent of variable j in term i.

For example, suppose that an input includes three predictor variables x1, x2, and x3 and the response variable y in the order x1, x2, x3, and y. Each row of T represents one term:

- [0 0 0 0] — Constant term or intercept
- [0 1 0 0] — x2; equivalently, x1^0 * x2^1 * x3^0
- [1 0 1 0] — x1*x3
- [2 0 0 0] — x1^2
- [0 1 2 0] — x2*(x3^2)

The `0` at the end of each term represents the response variable. In general, a column vector of zeros in a terms matrix represents the position of the response variable. If you have the predictor and response variables in a matrix and column vector, then you must include `0` for the response variable in the last column of each row.

Formula

A formula for model specification is a character vector or string scalar of the form `'y ~ terms'`.

- *`y`* is the response name.
- *`terms`* represents the predictor terms in a model using Wilkinson notation.

To represent predictor and response variables, use the variable names of the table input `tbl` or the variable names specified by using `VarNames`. The default value of `VarNames` is `{'x1','x2',...,'xn','y'}`.

For example:

- `'y ~ x1 + x2 + x3'` specifies a three-variable linear model with intercept.
- `'y ~ x1 + x2 + x3 – 1'` specifies a three-variable linear model without intercept. Note that formulas include a constant (intercept) term by default. To exclude a constant term from the model, you must include `–1` in the formula.

A formula includes a constant term unless you explicitly remove the term using `–1`.

Wilkinson Notation

Wilkinson notation describes the terms present in a model. The notation relates to the terms present in a model, not to the multipliers (coefficients) of those terms.

Wilkinson notation uses these symbols:

- `+` means include the next variable.
- `–` means do not include the next variable.
- `:` defines an interaction, which is a product of terms.
- `*` defines an interaction and all lower-order terms.
- `^` raises the predictor to a power, exactly as in `*` repeated, so `^` includes lower-order terms as well.
- `()` groups terms.

This table shows typical examples of Wilkinson notation.

Wilkinson Notation	**Terms in Standard Notation**
`1`	Constant (intercept) term
`x1^k`, where `k` is a positive integer	`x1`, $x1^2$, ..., $x1^k$
`x1 + x2`	`x1`, `x2`
`x1*x2`	`x1`, `x2`, `x1*x2`
`x1:x2`	`x1*x2` only
`–x2`	Do not include `x2`
`x1*x2 + x3`	`x1`, `x2`, `x3`, `x1*x2`
`x1 + x2 + x3 + x1:x2`	`x1`, `x2`, `x3`, `x1*x2`
`x1*x2*x3 – x1:x2:x3`	`x1`, `x2`, `x3`, `x1*x2`, `x1*x3`, `x2*x3`

Wilkinson Notation	Terms in Standard Notation
`x1*(x2 + x3)`	x1, x2, x3, x1*x2, x1*x3

For more details, see Wilkinson Notation.

Canonical Function

The default link function for a generalized linear model is the canonical link function.

Distribution	Canonical Link Function Name	Link Function	Mean (Inverse) Function
`'normal'`	`'identity'`	$f(\mu) = \mu$	$\mu = Xb$
`'binomial'`	`'logit'`	$f(\mu) = \log(\mu/(1 - \mu))$	$\mu = \exp(Xb) / (1 + \exp(Xb))$
`'poisson'`	`'log'`	$f(\mu) = \log(\mu)$	$\mu = \exp(Xb)$
`'gamma'`	`-1`	$f(\mu) = 1/\mu$	$\mu = 1/(Xb)$
`'inverse gaussian'`	`-2`	$f(\mu) = 1/\mu^2$	$\mu = (Xb)^{-1/2}$

Tips

- The generalized linear model `mdl` is a standard linear model unless you specify otherwise with the `Distribution` name-value pair.
- For other methods such as `devianceTest`, or properties of the `GeneralizedLinearModel` object, see `GeneralizedLinearModel`.
- After training a model, you can generate C/C++ code that predicts responses for new data. Generating C/C++ code requires MATLAB Coder™. For details, see Introduction to Code Generation.

Algorithms

- Stepwise regression is a systematic method for adding and removing terms from a linear or generalized linear model based on their statistical significance in explaining the response variable. The method begins with an initial model, specified using `modelspec`, and then compares the explanatory power of incrementally larger and smaller models.

  The `stepwiseglm` function uses forward and backward stepwise regression to determine a final model. At each step, the function searches for terms to add to the model or remove from the model based on the value of the `'Criterion'` name-value pair argument.

  The default value of `'Criterion'` for a linear regression model is `'sse'`. In this case, `stepwiselm` and `step` of `LinearModel` use the p-value of an F-statistic to test models with and without a potential term at each step. If a term is not currently in the model, the null hypothesis is that the term would have a zero coefficient if added to the model. If there is sufficient evidence to reject the null hypothesis, the function adds the term to the model. Conversely, if a term is currently in the model, the null hypothesis is that the term has a zero coefficient. If there is insufficient evidence to reject the null hypothesis, the function removes the term from the model.

  Stepwise regression takes these steps when `'Criterion'` is `'sse'`:

  1. Fit the initial model.

2. Examine a set of available terms not in the model. If any of the terms have p-values less than an entrance tolerance (that is, if it is unlikely a term would have a zero coefficient if added to the model), add the term with the smallest p-value and repeat this step; otherwise, go to step 3.
3. If any of the available terms in the model have p-values greater than an exit tolerance (that is, the hypothesis of a zero coefficient cannot be rejected), remove the term with the largest p-value and return to step 2; otherwise, end the process.

At any stage, the function will not add a higher-order term if the model does not also include all lower-order terms that are subsets of the higher-order term. For example, the function will not try to add the term `X1:X2^2` unless both `X1` and `X2^2` are already in the model. Similarly, the function will not remove lower-order terms that are subsets of higher-order terms that remain in the model. For example, the function will not try to remove `X1` or `X2^2` if `X1:X2^2` remains in the model.

The default value of `'Criterion'` for a generalized linear model is `'Deviance'`. `stepwiseglm` and `step` of `GeneralizedLinearModel` follow a similar procedure for adding or removing terms.

You can specify other criteria by using the `'Criterion'` name-value pair argument. For example, you can specify the change in the value of the Akaike information criterion, Bayesian information criterion, R-squared, or adjusted R-squared as the criterion to add or remove terms.

Depending on the terms included in the initial model, and the order in which the function adds and removes terms, the function might build different models from the same set of potential terms. The function terminates when no single step improves the model. However, a different initial model or a different sequence of steps does not guarantee a better fit. In this sense, stepwise models are locally optimal, but might not be globally optimal.

- `stepwiseglm` treats a categorical predictor as follows:
  - A model with a categorical predictor that has L levels (categories) includes $L - 1$ indicator variables. The model uses the first category as a reference level, so it does not include the indicator variable for the reference level. If the data type of the categorical predictor is `categorical`, then you can check the order of categories by using `categories` and reorder the categories by using `reordercats` to customize the reference level. For more details about creating indicator variables, see Automatic Creation of Dummy Variables.
  - `stepwiseglm` treats the group of $L - 1$ indicator variables as a single variable. If you want to treat the indicator variables as distinct predictor variables, create indicator variables manually by using `dummyvar`. Then use the indicator variables, except the one corresponding to the reference level of the categorical variable, when you fit a model. For the categorical predictor `X`, if you specify all columns of `dummyvar(X)` and an intercept term as predictors, then the design matrix becomes rank deficient.
  - Interaction terms between a continuous predictor and a categorical predictor with L levels consist of the element-wise product of the $L - 1$ indicator variables with the continuous predictor.
  - Interaction terms between two categorical predictors with L and M levels consist of the $(L - 1)*(M - 1)$ indicator variables to include all possible combinations of the two categorical predictor levels.
  - You cannot specify higher-order terms for a categorical predictor because the square of an indicator is equal to itself.

  Therefore, if `stepwiseglm` adds or removes a categorical predictor, the function actually adds or removes the group of indicator variables in one step. Similarly, if `stepwiseglm` adds or removes an interaction term with a categorical predictor, the function actually adds or removes the group of interaction terms including the categorical predictor.
- `stepwiseglm` considers `NaN`, `''` (empty character vector), `""` (empty string), `<missing>`, and `<undefined>` values in `tbl`, `X`, and `Y` to be missing values. `stepwiseglm` does not use observations with missing values in the fit. The `ObservationInfo` property of a fitted model indicates whether or

not `stepwiseglm` uses each observation in the fit.

Alternatives

- Use `fitglm` to create a model with a fixed specification. Use `step`, `addTerms`, or `removeTerms` to adjust a fitted model.

References

[1] Collett, D. Modeling Binary Data. New York: Chapman & Hall, 2002.

[2] Dobson, A. J. An Introduction to Generalized Linear Models. New York: Chapman & Hall, 1990.

[3] McCullagh, P., and J. A. Nelder. Generalized Linear Models. New York: Chapman & Hall, 1990.

See Also

`fitglm` | `GeneralizedLinearModel` | `predict`

Topics

Compare large and small stepwise models
Generalized Linear Models
Sequential Feature Selection

Introduced in R2013b