The Stata Journal (2017)
**17**, Number 1, pp. 240–249

# Generalized maximum entropy estimation of linear models

Paul Corral
The World Bank
Washington, DC
pcorralrodas@worldbank.org

Daniel Kuehn
Urban Institute
Washington, DC
dkuehn@urban.org

Ermengarde Jabir
American University
Washington, DC

**Abstract.** In this article, we describe the user-written `gmentropylinear` command, which implements the generalized maximum entropy estimation method for linear models. This is an information-theoretic procedure preferable to its maximum likelihood counterparts in many applications; it avoids making distributional assumptions, works well when the sample is small or covariates are highly correlated, and is more efficient than its maximum likelihood equivalent. We give a brief introduction to the generalized maximum entropy procedure, present the `gmentropylinear` command, and give an example using the command.

**Keywords:** st0473, gmentropylinear, generalized maximum entropy, maximum entropy, linear

## 1 Introduction

In this article, we introduce a user-written command, `gmentropylinear`, for fitting a linear model using generalized maximum entropy (GME) methods. GME estimation of the linear regression model provides an alternative to traditional estimation methods such as least-squares models or generalized linear models. Instead of minimizing squared residuals or maximizing a likelihood function defined by the researcher, maximum entropy methods select the linear model coefficients that are maximally informative, using an entropy measure of information content. Maximum entropy estimation selects the most conservative or noncommittal solution to the linear model. The GME linear model developed by Golan, Judge, and Miller (1996) builds on this maximum entropy principle by introducing stochastic moments into the optimization problem. GME provides a framework for fitting models that are robust to poor specification and to data that are partial or incomplete.

## 2 GME linear model

Using maximum entropy prevents the econometrician from imposing moment conditions that must be fulfilled on data that may be neither large nor well behaved.

The entropy measure used here is defined by Shannon (1948) as

$$H(p) \equiv -\sum_i p_i \ln p_i$$

 

where $p_i$ is the probability of observing outcome $i$. Any base for the logarithm will provide a viable entropy measure. Shannon (1948) used log base 2 for work with the communication of bits of information, although the natural logarithm is commonly used in econometric applications. When the full probabilistic distribution of a sample is known, the entropy measure equals zero. However, when the distribution is completely unknown, the probability distribution that imposes no priors on the data (but rather only on the information contained within the data) is the uniform distribution whose entropy measure equals the maximum value allowed, given by the entropy measure described above. Thus maximizing entropy based on the sample's information content leads to selecting the least-informed distribution.

In GME, the probabilities entering the entropy measure are weights associated with a vector of supports for both the parameters of the linear model (the $\beta$'s) and the error terms. The maximum entropy criterion from Jaynes (1957a,b) is used to select the set of probabilities, or weights, that is maximally informative but still consistent with the empirical data. Golan, Judge, and Miller's (1996) generalization of Jaynes's maximum entropy criterion to include stochastic moments of the data by additively including the entropy of the error term into the objective function is implemented here.

Linear regressions are commonly used in economics to model the relationship between a variable of interest and a set of explanatory variables. The GME approach loosens the assumptions of many alternative linear model estimation techniques, such as least squares, and imposes minimal distributional assumptions. Golan, Judge, and Perloff (1996) and Golan, Judge, and Miller (1996) have used Monte Carlo simulations to demonstrate that GME discrete choice and linear models provide more stable parameter estimates as collinearity of the covariates increases than their maximum likelihood or least-squares counterparts.

The linear GME model recovers probability distributions for the coefficients and the error terms. These probability distributions make use of the available sample information. The GME method for linear models is developed by Golan, Judge, and Miller (1996, 85), and the discussion below draws from that source.

To recover probability distributions, one must reparameterize the coefficients and the error terms. The generic version of the linear model to be fit is

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where $\mathbf{Y}$ is a $(T \times 1)$ matrix and $\mathbf{X}$ is a $(T \times K)$ matrix. The linear GME model reparameterizes both the $\boldsymbol{\beta}$ and the $\boldsymbol{\varepsilon}$ in the generic linear model as the expected values of a random variable defined on a probability distribution. Each coefficient in the GME framework has a bounded support space $\mathbf{z}_k$, associated with the $k$th variable, which is symmetrically built around zero and weighted by the vector $\mathbf{p}_k$ to reflect that the econometrician may not have prior knowledge to incorporate into the support space. Alternative support spaces not built around zero are allowable if, for example, the econometrician has prior knowledge of the value of $\boldsymbol{\beta}$. This would be similar in principle to the generalized cross entropy approach, where the probability weights are informed by prior knowledge rather than the supports. Both $\mathbf{z}_k$ and $\mathbf{p}_k$ are $(M \times 1)$

matrices. These weights must be in probability form; as such, their sum is equal to one (a requirement that will enter the system through the constraints). The coefficients are thus

$$\boldsymbol{\beta} = \mathbf{Zp} = \begin{bmatrix} \mathbf{z}_1' & \mathbf{0} & . & \mathbf{0} \\ \mathbf{0} & \boldsymbol{z}_2' & . & \mathbf{0} \\ . & . & . & . \\ \mathbf{0} & \mathbf{0} & . & \mathbf{z}_K' \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ . \\ \mathbf{p}_K \end{bmatrix} \tag{1}$$

where $\mathbf{Z}$ is a $(K \times KM)$ matrix and $\mathbf{p}$ is a weight vector of dimension $KM \times 1$. There are $M$ supports for each coefficient, and each support is associated with a probability weight. The support space recommended by Golan, Judge, and Miller (1996) is symmetric around zero and can be widened as needed to ensure that the true value for $\boldsymbol{\beta}$ lies inside the support space. Consequently, the product of these matrices is $(K \times 1)$, and for any particular $k$, the coefficient is equal to

$$\beta_k = \sum_m z_{km} p_{km}$$

Similarly, the error terms must also be parameterized as follows:

$$\boldsymbol{\varepsilon} \equiv \mathbf{Vw} = \begin{bmatrix} \mathbf{v}_1' & \mathbf{0} & . & \mathbf{0} \\ \mathbf{0} & \mathbf{v}_2' & . & \mathbf{0} \\ . & . & . & . \\ \mathbf{0} & \mathbf{0} & . & \mathbf{v}_T' \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ . \\ \mathbf{w}_T \end{bmatrix} \tag{2}$$

Thus an individual observation's error term is equal to

$$\varepsilon_t = \sum_j w_{tj} v_j$$

where $w_{tj}$ is the set of proper probabilities for each $t$ and $\mathbf{w}$ is the $(TJ \times 1)$ vectorization of $\mathbf{w}$. It is common to build the error support using the three-sigma rule $\mathbf{v} = (-3\widehat{\sigma}_Y, 0, 3\widehat{\sigma}_Y)$, where $\widehat{\sigma}_Y$ is the sample standard deviation for the dependent variable $\mathbf{Y}$ (Pukelsheim 1994). The linear model under the reparameterization done by Golan, Judge, and Miller (1996) becomes

$$\mathbf{Y} = \mathbf{XZp} + \mathbf{Vw}$$

The entropy term is maximized subject to the requirements of the proper probability distributions for $p_{km}$ and $w_{tj}$ and the $T$ information-moment constraints of the linear model (because $\mathbf{Y}$ is $(T \times 1)$, all $T$ information moments, or data points, enter through the constraints). Therefore, the Lagrangian is

$$\begin{aligned} \mathcal{L} = -\mathbf{p}' \ln \mathbf{p} - \mathbf{w}' \ln \mathbf{w} + \boldsymbol{\lambda}' (\mathbf{XZp} + \mathbf{Vw} - \mathbf{Y}) \\ + \boldsymbol{\delta}' \{\mathbf{1}_K - (\mathbf{I}_K \otimes \mathbf{1}_M') \mathbf{p}\} \\ + \boldsymbol{\gamma}' \{\mathbf{1}_T - (\mathbf{I}_T \otimes \mathbf{1}_J') \mathbf{w}\} \end{aligned}$$

where $\mathbf{1}_K$ is a $(K \times 1)$ vector of ones; the same holds for the other subscripts. The gradient of the Lagrangian is taken with respect to the Lagrangian parameters ($\boldsymbol{\lambda}$, $\boldsymbol{\delta}$, and $\boldsymbol{\gamma}$) and the probabilities ($\mathbf{p}$ and $\mathbf{w}$). The solutions for $\mathbf{p}$ and $\mathbf{w}$ are

$$\widehat{p}_{km} = \frac{\exp\left(-z_{km}\sum_t \widehat{\lambda}_t x_{tk}\right)}{\sum_m \exp\left(-z_{km}\sum_t \widehat{\lambda}_t x_{tk}\right)} \equiv \frac{\exp\left(-z_{km}\sum_t \widehat{\lambda}_t x_{tk}\right)}{\Omega_k\left(\widehat{\boldsymbol{\lambda}}\right)} \tag{3}$$

and

$$\widehat{w}_{tj} = \frac{\exp\left(-\widehat{\lambda}_t v_j\right)}{\sum_j \exp\left(-\widehat{\lambda}_t v_j\right)} \equiv \frac{\exp\left(-\widehat{\lambda}_t v_j\right)}{\Psi_t\left(\widehat{\boldsymbol{\lambda}}\right)} \tag{4}$$

The $\beta$'s and $\varepsilon$'s can be recovered by substituting $\mathbf{p}$ and $\mathbf{w}$ into (1) and (2). However, a more efficient way to solve for the $\beta$'s is to solve the unconstrained dual formulation of the problem, which is a function of the $\lambda$'s. The resulting function is referred to as the minimal value function by Golan, Judge, and Miller (1996).

$$\begin{aligned}\max_{\mathbf{p},\mathbf{w}} H(\mathbf{p},\mathbf{w}) &= \min_{\lambda}\left\{\sum_t y_t\lambda_t + \sum_k \ln\Omega_k(\boldsymbol{\lambda}) + \sum_t \ln\Psi_t(\boldsymbol{\lambda})\right\} \\ &= \min_{\lambda}\left[\sum_t y_t\lambda_t + \sum_k \ln\left\{\sum_m \exp\left(-z_{km}\sum_t \lambda_t x_{tk}\right)\right\}\right] \\ &\quad + \sum_t \ln\left\{\sum_j \exp\left(-\lambda_t v_j\right)\right\}\end{aligned}$$

Minimizing with respect to $\lambda$ solves the dual formulation and provides the optimal $\lambda$'s, which are subsequently used to solve for the optimal $\mathbf{p}$'s and $\mathbf{w}$'s [(3) and (4)]. These, in turn, are the weights associated with the support space, which generates the parameters and the errors. Although this approach appears more roundabout, it offers greater efficiencies because there are only $T$ $\lambda$'s to estimate rather than a combination of $KM$ $p$'s and $TJ$ $w$'s. The `gmentropylinear` command optimizes the dual unconstrained model using the Newton–Raphson method, following the implementation in Cameron and Trivedi (2010). All analyses conducted by the authors have rapidly converged using this procedure.

## 2.1 Asymptotic variance of the GME estimator

Mittelhammer, Cardell, and Marsh (2013) develop the asymptotic theory and inference for the linear GME estimator.[1] The authors state that the GME estimator is asymptotically and normally distributed with a variance–covariance matrix given by

1. See Mittelhammer, Judge, and Miller (2000) for a full derivation of the asymptotic properties of the model.

$$\widehat{\text{Var}}\left(\widehat{\boldsymbol{\beta}}\right) = \frac{\widehat{\sigma}^2_{\lambda}\left(\widehat{\boldsymbol{\beta}}\right)}{\zeta^2\left(\widehat{\boldsymbol{\beta}}\right)}\left(\mathbf{X}'\mathbf{X}\right)^{-1}$$

where

$$\widehat{\sigma}^2_{\lambda}\left(\widehat{\boldsymbol{\beta}}\right) = \frac{1}{T}\sum_{t=1}^{T}\lambda_t^2$$

and

$$\zeta^2\left(\widehat{\boldsymbol{\beta}}\right) = \left[\frac{1}{T}\sum_{t=1}^{T}\left\{\sum_{j=1}^{J}v_j^2 w_{tj} - \left(\sum_{j=1}^{J}v_j w_{tj}\right)^2\right\}^{-1}\right]^2$$

The variance of the estimators, because it is asymptotically normally distributed, can be used for hypothesis testing on the values of $\boldsymbol{\beta}$.

## 3 The gmentropylinear command

This command fits a linear model using the GME principle. The syntax reflects standard linear regression syntax in Stata, with one exception. Unlike the discrete choice version of the GME model, `gmentropylogit` (Corral and Terbish 2015), users must provide the parameter support space for the $\beta$'s in the `gmentropylinear` command. The support space for the error terms is set by default to a dimension of three, and it is equally and symmetrically built around zero using the three-sigma rule as suggested by Golan, Judge, and Miller (1996). However, the command allows the user to specify an alternative error support space in both dimension and values. The $\boldsymbol{\beta}$ coefficient support space is a $K \times M$ matrix, where $K$ is the number of covariates including the constant (estimation without the constant term is also allowed) and $M$ is the number of supports. Although some packages for the GME linear model in other software build parameter support spaces around ordinary least-squares (OLS) estimates of the model, we recommend a support space that is symmetric around zero instead. When sample sizes are small, and a wide support space around zero is built, the GME estimator is usually a better estimator than the OLS (Mittelhammer, Cardell, and Marsh 2013). Golan, Judge, and Miller (1996, 109) note that although the GME solution is consistent, it is still likely to suffer from small-sample bias. Nevertheless, in simulations, it still has lower mean square errors than traditional methods. If the user has prior knowledge about the true $\beta$'s, a narrow support around this prior knowledge will provide a far better fit than its OLS counterpart, particularly in small sample sizes (Mittelhammer, Cardell, and Marsh 2013). Thus the GME's quality is dependent upon the supports chosen. Nevertheless, wide supports built around zero usually outperform their OLS counterparts.

One of GME's principal advantages is that it provides the solution least committed to potentially wrong specification assumptions (aside from linearity). Building the

support space from potentially mistaken prior OLS estimates will push the GME result toward the OLS result by adding information from the OLS specification into the entropy maximization problem. As Golan, Judge, and Miller (1996) note, a wider support space increases the impact of the data, while a tighter support space increases the impact of the support. The support space is generated simply by defining an appropriate matrix:

`matrix` *matname* `=` *matrix_expression*

Once a matrix for the $\beta$ support has been specified, the user can proceed with the command:

`gmentropylinear` *depvar* [ *indepvars* ] [ *if* ] [ *in* ], `support(`*matrix*`)`
[ `sigmavalue(#) endpoint(#) lambda(`*string*`) wmat(`*string*`) residual(`*string*`)`
`nosigma nocons` ]

The `support(`*matrix*`)` component of the command identifies the matrix defining the support space for the coefficients and must be provided by the user as a matrix. `support()` is required. The `gmentropylinear` command provides various options. Although the specification of the coefficient support is required, the support for the error terms is constructed by the command. However, the user can modify the number of support spaces and decide whether to use the three-sigma rule. The `endpoint(#)` option tells the `gmentropylinear` command how many supports are used in the estimation for the error term. The default for the error supports is to use the empirical three-sigma rule, with $J = 3$. Thus the default error space for each observation is $v = (-3\widehat{\sigma}_Y, 0, 3\widehat{\sigma}_Y)$. The `sigmavalue()` option is used to specify the sigma endpoint; the default is `sigmavalue(3)`. If the user specifies `sigmavalue(4)`, then $v = (-4\widehat{\sigma}_Y, 0, 4\widehat{\sigma}_Y)$. The use of the empirical $\sigma$ can also be overridden with the `nosigma` option. Therefore, if the user specifies `sigmavalue(4)` and `nosigma`, then $v = (-4, 0, 4)$. Finally, the user can adjust the number of supports ($J$). This is done with the `endpoint()` option; the default is `endpoint(3)`. Regardless of the number of supports specified, the command will always adjust this to be odd numbered.

Several options return estimates from the model, including `lambda()`, which returns the estimated $\lambda$'s, `wmat()`, which returns the estimated `w`, and `residual()`, which returns the residuals. One can also suppress the constant term with the `nocons` option.

Besides the estimates of the coefficients and their standard errors, `gmentropylinear` provides the final entropy for the model as well as the normalized signal entropy:

$$S(\widehat{p}) = \frac{-\sum_k \sum_j p_{kj} \ln p_{kj}}{K \ln M}$$

The normalized entropy for the noise parameters is also included as follows:

$$S(\widehat{w}) = \frac{-\sum_t \sum_j w_{tj} \ln w_{tj}}{T \ln J}$$

The pseudo-$R^2$, measuring goodness of fit within the sample analyzed, follows from the following normalized entropy metric:

$$\text{Pseudo-}R^2 = 1 - S\left(\widehat{p}\right)$$

The normalized entropy measure scales the estimated entropy by the maximum possible entropy of a problem with the provided number of variables, elements of the support space, and observations. The maximum possible entropy occurs only when the distribution of probabilities over the $p_m$'s and the $w_j$'s is uniform and departs from this maximum entropy point only when the moment constraints are included (Soofi 1992). The data, which enter through the moment constraints, add information that reduces uncertainty and pushes the solution away from the uniform distribution. As its name suggests, the normalized entropy figure can be compared across analyses and ranges between zero and one.

# 4 Example

The following example uses the `gmentropylinear` command to estimate the (logged) price of a car in Stata's `auto.dta`. Price is estimated here as a function of miles per gallon, weight, and whether the car is foreign. First, the parameter support matrix (`support()`) is defined. This is a $(K \times M)$ matrix, where $K = 4$, because there are three coefficients and a constant term to estimate. In this case, $M = 5$, although other dimensions are admissible. Running the estimation without a constant is also possible with the `nocons` option.

```
. sysuse auto
(1978 Automobile Data)
. generate lnprice=ln(price)
. matrix support=(-1,-.5,0,.5,1)\(-1,-.5,0,.5,1)\(-5,-2.5,0,2.5,5)\
> (-5,-2.5,0,2.5,5)
. gmentropylinear lnprice mpg weight foreign, support(support) sigmavalue(3)
> endpoint(3) wmat(err) residual(error) lambda(lambda)
Iteration 1:    Entropy = 87.735061
Iteration 2:    Entropy = 82.8903845
Iteration 3:    Entropy = 82.3891972
Iteration 4:    Entropy = 82.3207142
Iteration 5:    Entropy = 82.3199698
Iteration 6:    Entropy = 82.3199698
Iteration 7:    Entropy = 82.3199698
Generalized Maximum Entropy (Linear)              Number of obs      =     74
                                                  Degrees of freedom =      3
                                                  Model Entropy      =   82.3
                                                  Pseudo R2          = 0.2133
                                                  Signal entropy     = 0.7867
                                                  Noise entropy      = 0.9503
```

| lnprice | Coef. | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| mpg | .0455501 | .0107623 | 4.23 | 0.000 | .0244564 | .0666437 |
| weight | .0008686 | .0000915 | 9.50 | 0.000 | .0006893 | .0010478 |
| foreign | .7168477 | .0991786 | 7.23 | 0.000 | .5224612 | .9112342 |
| _cons | 4.824633 | .4896719 | 9.85 | 0.000 | 3.864894 | 5.784372 |

Finally, `gmentropylinear` stores the following in `e()`:

Scalars

| | |
|---|---|
| `e(N)` | number of observations |
| `e(df_m)` | model degrees of freedom |
| `e(entropy)` | final entropy for the model |
| `e(int_entropy)` | initial entropy |
| `e(pseudoR2)` | pseudo-$R^2$ |
| `e(sign_entropy)` | normalized entropy for the signal |
| `e(noise_entropy)` | normalized entropy for the noise |

Macros

| | |
|---|---|
| `e(depvar)` | name of dependent variable |
| `e(properties)` | `b V` |

Matrices

| | |
|---|---|
| `e(b)` | coefficient vector |
| `e(V)` | variance–covariance matrix of the estimators |
| `e(esupport)` | error support space specified |
| `e(betaprobs)` | coefficient parameter support space |

Functions

| | |
|---|---|
| `e(sample)` | marks estimation sample |

## 5 Conclusion

In this article, we described the user-written `gmentropylinear` command, which provides users with a GME alternative for fitting linear models. GME estimation is particularly advantageous when estimation is performed on small datasets or when the user is unsure about the appropriate model specification and would like to find the maximally noncommittal solution. The GME approach shrinks the joint entropy distance between the data being analyzed and an assumption of uniform priors where there is complete uncertainty about the underlying distribution. Implementing a dual unconstrained model allows for equal emphasis to be placed on the precision of the estimates as well as on prediction, where the estimated probabilities yield the distribution of the parameters of interest (the error and coefficient estimates) up to the $M - 1$ moment for the coefficient and the $J - 1$ moment for the error term (Golan, Judge, and Perloff 1996). This approach also benefits from being robust to collinearity. As with other robust estimators, discrete support spaces for the parameters (coefficients and errors) must be specified. The command outlined here generates GME estimates derived from user-specified parameter supports and error supports that are provided either by the user or by the defaults.

## 6 Acknowledgments

The code presented in this article was written with the support and encouragement of Amos Golan and is part of the examples used in his upcoming book on information theory. Additionally, we would like to thank Minh Nguyen for comments provided on the code. Finally, we thank the reviewer.

Any mistake or omission is the authors' responsibility alone.

## 7 References

Cameron, A. C., and P. K. Trivedi. 2010. *Microeconometrics Using Stata*. Rev. ed. College Station, TX: Stata Press.

Corral, P., and M. Terbish. 2015. Generalized maximum entropy estimation of discrete choice models. *Stata Journal* 15: 512–522.

Golan, A., G. Judge, and J. M. Perloff. 1996. A maximum entropy approach to recovering information from multinomial response data. *Journal of the American Statistical Association* 91: 841–853.

Golan, A., G. G. Judge, and D. Miller. 1996. *Maximum Entropy Econometrics: Robust Estimation with Limited Data*. Chichester, UK: Wiley.

Jaynes, E. T. 1957a. Information theory and statistical mechanics. *Physical Review* 106: 620–630.

———. 1957b. Information theory and statistical mechanics. II. *Physical Review* 108: 171–190.

Mittelhammer, R., N. S. Cardell, and T. L. Marsh. 2013. The data-constrained generalized maximum entropy estimator of the GLM: Asymptotic theory and inference. *Entropy* 15: 1756–1775.

Mittelhammer, R. C., G. G. Judge, and D. J. Miller. 2000. *Econometric Foundations*. Cambridge: Cambridge University Press.

Pukelsheim, F. 1994. The three sigma rule. *American Statistician* 48: 88–91.

Shannon, C. E. 1948. A mathematical theory of communication. *Bell System Technical Journal* 27: 379–423.

Soofi, E. S. 1992. A generalizable formulation of conditional logit with diagnostics. *Journal of the American Statistical Association* 87: 812–816.

**About the authors**

Paul Corral is a data scientist in the Poverty and Equity Global Practice at the World Bank.

Daniel Kuehn is a research associate in the Income and Benefits Policy Center at the Urban Institute.

Ermengarde Jabir is an economics PhD candidate at American University.