MaxEnt

A program for maximum entropy modelling of species geographic distributions, written by Steven Phillips, Miro Dudik and Rob Schapire, with support from AT&T Labs-Research, Princeton University, and the Center for Biodiversity and Conservation, American Museum of Natural History. Thank you to the authors of the following free software packages which we have used here: ptolemy/plot, gui/layouts, gnu/getopt and com/mindprod/ledatastream.

This page contains reference information for the MaxEnt program. Background information on the method can be found in the following two papers:

Steven J. Phillips, Robert P. Anderson, Robert E. Schapire.
Maximum entropy modeling of species geographic distributions.
Ecological Modelling, Vol 190/3-4 pp 231-259, 2006.

Steven J. Phillips, Miroslav Dudik.
Modeling of species distributions with Maxent: new extensions and a comprehensive evaluation.
Ecography, Vol 31 pp 161-175, 2008.

The model for a species is determined from a set of environmental or climate layers (or "coverages") for a set of grid cells in a landscape, together with a set of sample locations where the species has been observed. The model expresses the suitability of each grid cell as a function of the environmental variables at that grid cell. A high value of the function at a particular grid cell indicates that the grid cell is predicted to have suitable conditions for that species. The computed model is a probability distribution over all the grid cells. The distribution chosen is the one that has maximum entropy subject to some constraints: it must have the same expectation for each feature (derived from the environmental layers) as the average over sample locations.

Inputs, Outputs and Parameters

Input files, output directory and algorithm parameters can be specified through the user interface, or on a command line. The user interface is best for doing single runs, while the command line is useful for repeated runs or automatically performing a sequence of runs with variations in the set of inputs.

Inputs:

- **Samples.** Given by a file in comma-separated value format. The first line is a header line, while later lines have the format: species, longitude, latitude. For example

```
Species, Long, Lat
Blue-headed Vireo, -89.9, 48.6
Loggerhead Shrike, -87.15, 34.95
...
```

Any number of species can be represented in the same file. Individual species can be selected or deselected before starting a run, and only selected species will be modeled.

- **Environmental layers.** Given by a directory containing the layers. The layers must either be in ESRI ASCII grid format (described below), with filenames ending in ".asc", or Diva-GIS grid format, with filenames ending in ".grd" and ".gri". By default, all layers in the directory are used in the modeling, but individual layers can be deselected before starting a run. Each layer can be continuous (having real or integer values) or categorical (having a small number of discrete values). The environmental layers can also be given in a SWD format file as described next.
- **SWD (samples-with-data) format.** You can give the samples values for the environmental variables directly in the .csv file, as in the following example:

```
Species, X, Y, Var1, Var2, Var3

Blue-headed Vireo, 310186, 8243704, 1, 19.5, 0.91

Blue-headed Vireo, 300243, 8173341, 2, 18.3, 1.04

Loggerhead Shrike, 290434, 8192276, 4, 20.7, 0.88
```

This file is then used as the sample file. The value -9999 is interpreted as NODATA, and should be used if some samples are lacking data for some of the environmental variables. The "X" and "Y" fields are for geographic coordinates, though they are not used by the MaxEnt program if all environmental data is given in the SWD format file. In a similar way, a set of background points can also be given environmental data, using the same format, for example:

```
Species, X, Y, Var1, Var2, Var3

background, 320268, 8428840, 1, 17.5, 0.55

background, 301886, 8432739, 2, 18.1, 0.65
```

The SWD format file with background data is then used in place of the environmental layers directory. For background data, the "Species" column is ignored (we've used "background" for clarity only), as are any lines containing NODATA values. The two formats can be mixed: samples can be specified in SWD format, with background data given in grids. However, if background data is given in SWD format, then the samples must be too.

- **Projection directory.** An optional directory (or SWD format file) containing a second set of environmental layers. The layers must have the same names as those in the "Environmental layers" directory, though they might describe a different geographic area. The projection process is described below.

Algorithm Parameters:

- **Feature types.** The environmental layers are used to produce "features", which constrain the probability distribution that is being computed. The available feature types are linear, quadratic, product, threshold and discrete. Using "auto features" allows the set of features used to depend on the number of presence records for the species being modeled, using general empirically-derived rules.
    - Linear features constrain the output distribution for each species to have the same expectation of each of the continuous environmental variables as the sample locations for that species. A linear feature is simply one of the continuous environmental variables.
    - Quadratic features (when used together with linear features) constrain the output distribution to have the same expectation and variance of the environmental variables as the samples. A quadratic feature is the square of one of the continuous environmental variables.
    - A product feature is the product of two continuous environmental variables; when used with linear and quadratic features, product features constrain the output distributions to have the same covariance for each pair of environmental variables as the samples.
    - A threshold feature is derived from a continuous environmental variable. For a threshold value *v*, the threshold feature is binary (taking values 0 and 1) and is 1 when the variable has value greater than *v*. The effect of a threshold feature is to make the total probability of grid cells with a value greater than the threshold be equal to the fraction of sample locations with the value above the threshold.
    - A hinge feature is also derived from a continuous environmental varaible. It is like a linear feature, but it is constant below a threshold *v*.
    - Discrete features are automatically made for each selected categorical variable. One feature is made for each possible value of each categorical variable: the feature for a value *v* is binary (taking values 0 and 1) and is 1 when the variable has value *v*. The effect of a discrete feature is to make the total probability of grid cells with a particular value of the categorical variable be equal to the fraction of sample locations with that value.
- **Control parameters.** There are a number of control parameters available, either on the main interface or the "Settings" panel. A tooltip (little text description) appears if you point the mouse at a control parameter, describing its effect.

Outputs:

All output files are written in the *output directory*. The summary of a maxent run is given in

- *maxentResults.csv*
  listing the number of training samples used for learning, values of training gain and test gain and AUC. Test gain and AUC are given only when a test sample file is provided or when a specified percentage of the samples is set aside for testing. If a jackknife is performed, the regularized training gain and (optionally) test gain and AUC for each part of the jackknife are included here.

- *maxent.log*
  records the parameters and options chosen for the run, and some details of the run that are useful for troubleshooting.

In addition, maxent produces several files for every species. For a species called *mySpecies*, it produces files

- *mySpecies.html*
  the main output file, containing statistical analyses, plots, pictures of the model, and links to other files. It also documents parameter and control settings that were used to do the run.
- *mySpecies.asc* (or *mySpecies.grd*)
  containing the probabilities in ESRI ASCII grid format (or in DIVA-Gis format if -H switch is used)
- *mySpecies.lambdas*
  containing the computed values of the constants *c1, c2, ...* (described below)
- *mySpecies.png*
  is a picture of the prediction
- *mySpecies_omission.csv*
  describing the predicted area and training and (optionally) test omission for various raw and cumulative thresholds
- various plots for jackknifing and response curves, in the *plots* subdirectory.

The *output format* for predicted distributions is either *raw,*, *logistic* (the default) or *cumulative*. For raw output, the output values are probabilities (between 0 and 1) such that the sum over all cells used during training is 1. Typical values are therefore extremely small. For logistic output, the values are again probabilities (between 0 or 1), but scaled up in a non-linear way for easier interpretation. If typical presences used during training are from environmental conditions where probability of presence is around 0.5, then the logistic output can be interpreted as predicted probability of presence (otherwise they can be interpreted as relative suitability). If $p(x)$ is the raw output for environmental conditions x, the corresponding logistic value is $c\, p(x) / (1 + c\, p(x))$ for a particular value of c (namely, the exponential of the entropy of the raw distribution). For the cumulative output format, the value at a grid cell is the sum of the probabilities of all grid cells with no higher probability than the grid cell, times 100. For example, the grid cell that is predicted as having the best conditions for the species, according to the model, will have cumulative value 100, while cumulative values close to 0 indicate predictions of unsuitable conditions.

ESRI ASCII Grid Format

(Copied from the ArcWorkstation 8.3 Help File)

The ASCII file must consist of header information containing a set of keywords, followed by cell values in row-major order. The file format is

```
<NCOLS xxx>
<NROWS xxx>
```

```
  <XLLCENTER xxx | XLLCORNER xxx>
  <YLLCENTER xxx | YLLCORNER xxx>
  <CELLSIZE xxx>
  {NODATA_VALUE xxx}
  row 1
  row 2
  ...
  row n
```

where `xxx` is a number, and the keyword `nodata_value` is optional and defaults to -9999. Row 1 of the data is at the top of the grid, row 2 is just under row 1 and so on. For example:

```
  ncols         386
  nrows         286
  xllcorner     -128.66338
  yllcorner     13.7502065
  cellsize      0.2
  NODATA_value  -9999
  -9999 -9999 -123 -123 -123 -9999 -9999 -9999 -9999 -9999 ...
  -9999 -9999 -123 -123 -123 -9999 -9999 -9999 -9999 -9999 ...
  -9999 -9999 -117 -117 -117 -119 -119 -119 -119 -119 -9999 ...
  ...
```

The `nodata_value` is the value in the ASCII file to be assigned to those cells whose true value is unknown. Cell values should be delimited by spaces. No carriage returns are necessary at the end of each row in the grid. The number of columns in the header is used to determine when a new row begins. The number of cell values must be equal to the number of rows times the number of columns.

The current implementation of maxent requires fields `xllcorner`, `yllcorner`*and* `nodata_value`.

How it works

This is a very brief description -- for more details, please see the papers described above. Here we first describe an unregularized version (with the regularization value set to zero); in practice, we always recommend to use regularization. Without regularization, the distribution being computed is the one that has maximum entropy among those satisfying the constraints that the expectation of each feature matches its empirical average. This distribution can be proved to be the same as the Gibbs distribution that maximizes the product of the probabilities of the sample locations, where a Gibbs distribution takes the form

$$P(x) = \exp(c1 * f1(x) + c2 * f2(x) + c3 * f3(x) ...) / Z$$

Here $c1$, $c2$, ... are constants, $f1$, $f2$, ... are the features, and Z is a scaling constant that ensures that P sums to 1 over all grid cells. The algorithm that is implemented by this program is guaranteed to converge to values of $c1$, $c2$, ..., that give the (unique) optimum distribution P.

For each species, the program starts with a uniform distribution, and performs a number of iterations, each of which increases the probability of the sample locations for the species. The probability is displayed in terms of "gain", which is the log of the number of grid cells minus the log loss (average of the negative log probabilities of the sample locations). The gain starts at zero (the gain of the uniform distribution), and increases as the program increases the probabilities of the sample locations. The gain increases iteration by iteration, until the change from one iteration to the next falls below the *convergence threshold*, or until *maximum iterations* have been performed.

In the regularized case, the gain is lower by an additional term, which is the weighted sum of the absolute values of *c1*, *c2*, ... This limits overfitting and prevents *c1*, *c2*, ... from becoming arbitrarily large. Minimizing the regularized loss (or equivalently, maximizing the regularized gain) corresponds to maximizing the entropy of the distribution subject to a relaxed constraint that feature expectations be only close to feature averages over sample locations rather than exactly equal to them.

Regularization and feature class selection

The predictive performance of the MaxEnt is influenced by the choice of feature types and the regularization constants. Here we describe the default settings, which can be overridden, if desired, using the command line flags described below. By default (i.e., when using "Auto features"), all feature types are used when there are at least 80 training samples; from 15 to 79 samples, linear, quadratic and hinge features are used; from 10 to 14 samples, linear and quadratic features are used; below 10 samples, only linear features are used.

The default values for the constants c1, c2 described above is an empirically tuned value (called "beta", and depending on the feature type and the number of samples) divided by the square root of the number of samples. The default values for beta for the various feature types are given in the following tables, with interpolation in between:

Linear (2-9 samples)

Sample size	0	10	30	100+
Beta	1.0	1.0	0.2	0.05

Linear + Quadratic (10-79 samples)

Sample size	0	10	17	30	100+
Beta	1.3	0.8	0.5	0.25	0.05

Linear + Quadratic + Product (80+ samples)

Sample size	0	10	17	30	100+
Beta	2.6	1.6	0.9	0.55	0.05

Threshold (80+ samples)

Sample size	0	100+
Beta	2.0	1.0

Hinge (15+ samples)

Sample size	0+
Beta	0.5

Categorical (15+ samples)

Sample size	0+	10	17+
Beta	0.65	0.5	0.25

Projections

The values of *c1*, *c2*, ... and *Z* that were computed for features derived from the "Environmental layers" are used to compute weights using the layers in the "Projection directory". Note that these weights are not probabilities and they need not sum to one since they use the normalization constant computed for "Environmental layers" rather than the one for "Projection directory". Their relative magnitudes represent how much a given locale is favored by the species over another locale. For each species, the weights are written in a file *mySpecies_<dir>.asc* in the output directory, where <dir> is the name of the projection directory. By default, two kinds of "clamping" are done during the projection process. First, the environmental layers are clamped: if a layer in the projection directory has values that are greater than the maximum of the corresponding layer used during training, those values are reduced to the maximum, and similarly for values below the corresponding minimum. Second, features are also clamped: if a feature derived from the projection layers has value greater than its maximum on the training data, it is reduced to the maximum, and similarly for values below the corresponding minimum. Both forms of clamping help to alleviate problems that can arise from making predictions outside the range of data used in training the model.

Background Points

As described above, the maxent distribution is calculated over the set of pixels that have data for all environmental variables. However, if the number of pixels is very large, processing time increases without a significant improvement in modeling performance. For that reason, when the number of pixels with data is larger than 10,000 a random sample of 10,000 "background" pixels is used to represent the variety of environmental conditions present in the data. The maxent distribution is then computed over the union of the "background" pixels and the samples for the species being modeled. The number 10,000 can be changed from the "Settings" panel, or by using a command-line flag: see the batch-mode section below.

Memory Issues

If the environmental layers are very large files, you may get "out of memory" or "heap space" errors when you try to run the program. There are a number of ways to address this problem.

- First, make sure that you are clicking on the maxent.bat file, rather than the maxent.jar file.
- Second, make sure that java is being given close to the maximum memory available on your computer. The maxent.bat file (or any command-line invocation) should begin "java -mxXXXm", where XXX is a little less than the number of megabytes of memory in your computer (e.g., use the flag "-mx900m" if you have a gigabyte of memory). It shouldn't equal the amount of memory in your computer, otherwise "thrashing" will occur as the last of the memory is consumed. An exception is on Microsoft Windows systems with multiple gigabytes of memory: Windows cannot give java the large contiguous block of memory it desires, so unfortunately you are limited to a maximum of about 1.3 gigabytes.
- Third, you can create SWD-format files (described above) containing the environmental conditions at the sample localities and a random set of background pixels (for example, using a GIS) so that the maxent program doesn't need to load the large environmental layers files. If you provide the input in this format, you'll probably want to project the resulting model onto your original environmental layers, so you should give their location in the "projection directory". The projection process is memory-efficient, as it doesn't need to hold all the environmental variables in memory at the same time.

Format of the lambda file

The coefficients of the Maxent model for a species are output in a file called *species*.lambdas. The entries in the lambdas file are lines of the form: *feature, lambda, min, max*. The exponent of the Maxent model is calculated as

exponent = lambda_1 * $(f_1(x)\text{-min}_1)/(\text{max}_1 - \text{min}_1)$ + ... + lambda_n * $(f_n(x)\text{-min}_n)/(\text{max}_n\ \text{-min}_n)$ - linearPredictorNormalizer

In other words, features are scaled so that their values would lie between 0 and 1 on the training data. By default, all features are clamped prior to projection of the model onto new data - see section "Projections" above. The linearPredictorNormalizer is a constant chosen so that the exponent is always non-positive (for numerical stability). Terms corresponding to hinge features are evaluated slightly differently. For example, the hinge feature prec' derived from the layer prec and described by the line: *prec', lambda, min, max* evaluates to the term

lambda * clamp_at_0(prec-min)/(max-min)

i.e., if prec< min then the value is 0 otherwise it is (prec-min)/(max-min). For the reverse hinge feature *prec`, lambda, min, max*, the term is

lambda * clamp_at_0(max-prec)/(max-min)

The densityNormalizer is the normalization term Z calculated over the background. The Maxent raw output is therefore:

raw = exp(sum lambda_i * (f_i(x)-min_i)/(max_i - min_i) - linearPredictorNormalizer) / densityNormalizer

Lastly, logistic output is calculated using the entropy given at the end of the lambda file: logistic = raw * exp(entropy) / (1 + raw * exp(entropy)).

Batch mode

All parts of the interface can be set from the command line, and the Run button can be automatically pressed after startup. This allows for the program to be invoked in batch mode, multiple times in sequence, if required. The command line flags can also be added to the maxent.bat file, at the end of the "java ..." line, to change the default settings of the program. Some of the more common flags have abbreviations that can be used instead of the full flag. As an example, the following two invocations are equivalent:

```
java -mx512m -jar maxent.jar environmentallayers=layers samplesfile=samples\bradypus.csv outputdirectory=outputs togglelayertype=ecoreg redoifexists autorun
```

```
java -mx512m -jar maxent.jar -e layers -s samples\bradypus.csv -o outputs -t ecoreg -r -a
```

Any boolean flag can be given the prefix "no" or "dont" to turn the flag off. Abbreviations for boolean flags toggle the default value. The available flags are, in no particular order:

Flag	Abbrv	Type	Default	Meaning
responsecurves	P	boolean	false	Create graphs showing how predicted relative probability of occurrence depends on the value of each environmental variable
pictures		boolean	true	Create a .png image for each output grid

jackknife	J	boolean	false	Measure importance of each environmental variable by training with each environmental variable first omitted, then used in isolation
outputformat		string	cloglog	Representation of probabilities used in writing output grids. See Help for details
outputfiletype		string	asc	File format used for writing output grids
outputdirectory	o	directory		Directory where outputs will be written. This should be different from the environmental layers directory.
projectionlayers	j	file/directory		Location of an alternate set of environmental variables. Maxent models will be projected onto these variables. Can be a .csv file (in SWD format) or a directory containing one file per variable. Multiple projection files/directories can be separated by commas.
samplesfile	s	file		Please enter the name of a file containing presence locations for one or more species.
environmentallayers	e	file/directory		Environmental variables can be in a directory containing one file per variable, or all together in a .csv file in SWD format. Please enter a directory name or file name.

randomseed		boolean	false	If selected, a different random seed will be used for each run, so a different random test/train partition will be made and a different random subset of the background will be used, if applicable.
logscale		boolean	true	If selected, all pictures of models will use a logarithmic scale for color-coding.
warnings		boolean	true	Pop up windows to warn about potential problems with input data. Regardless of this setting, warnings are always printed to the log file.
tooltips		boolean	true	Show messages that explain various parts of the interface, like this message
askoverwrite	r	boolean	true	If output files already exist for a species being modeled, pop up a window asking whether to overwrite or skip. Default is to overwrite.
skipifexists	S	boolean	false	If output files already exist for a species being modeled, skip the species without remaking the model.
removeduplicates		boolean	true	Remove duplicate presence records. If environmental data are in grids, duplicates are records in the same grid cell. Otherwise, duplicates are records with identical coordinates.

writeclampgrid		boolean	true	Write a grid that shows the spatial distribution of clamping. At each point, the value is the absolute difference between prediction values with and without clamping.
writemess		boolean	true	A multidimensional environmental similarity surface (MESS) shows where novel climate conditions exist in the projection layers. The analysis shows both the degree of novelness and the variable that is most out of range at each point.
randomtestpoints	X	integer	0	Percentage of presence localities to be randomly set aside as test points, used to compute AUC, omission etc.
betamultiplier	b	double	1.0	Multiply all automatic regularization parameters by this number. A higher number gives a more spread-out distribution.
maximumbackground	MB	integer	10000	If the number of background points / grid cells is larger than this number, then this number of cells is chosen randomly for background points
biasfile		file		Sampling is assumed to be biased according to the sampling distribution given in this grid file. Values in this file must not be zero or negative. MaxEnt will factor out the bias. Requires environmental

				data to be in grids, rather than a SWD format file
testsamplesfile	T	file		Use the presence localities in this file to compute statistics (AUC, omission etc.) The file can contain different localities for different species. It takes precedence over the random test percentage.
replicates		integer	1	Number of replicate runs to do when cross-validating, bootstrapping or doing sampling with replacement runs
replicatetype		string	crossvalidate	If replicates > 1, do multiple runs of this type: Crossvalidate: samples divided into *replicates* folds; each fold in turn used for test data. Bootstrap: replicate sample sets chosen by sampling with replacement. Subsample: replicate sample sets chosen by removing *random test percentage* without replacement to be used for evaluation.
perspeciesresults		boolean	false	Write separate maxentResults file for each species
writebackgroundpredictions		boolean	false	Write .csv file with predictions at background points
responsecurvesexponent		boolean	false	Instead of showing the logistic value for the y axis in response curves, show the exponent (a linear combination of features)

linear	l	boolean	true	Allow linear features to be used
quadratic	q	boolean	true	Allow quadratic features to be used
product	p	boolean	true	Allow product features to be used
threshold		boolean	false	Allow threshold features to be used
hinge	h	boolean	true	Allow hinge features to be used
addsamplestobackground	d	boolean	true	Add to the background any sample for which has a combination of environmental values that isn't already present in the background
addallsamplestobackground		boolean	false	Add all samples to the background, even if they have combinations of environmental values that are already present in the background
autorun	a	boolean	false	Start running as soon as the the program starts up
writeplotdata		boolean	false	Write output files containing the data used to make response curves, for import into external plotting software
fadebyclamping		boolean	false	Reduce prediction at each point in projections by the difference between clamped and non-clamped output at that point
extrapolate		boolean	true	Predict to regions of environmental space outside the limits encountered during training
visible	z	boolean	true	Make the Maxent user interface visible

autofeature	A	boolean	true	Automatically select which feature classes to use, based on number of training samples
doclamp		boolean	true	Apply clamping when projecting
outputgrids	x	boolean	true	Write output grids. Turning this off when doing replicate runs causes only the summary grids (average, std deviation etc.) to be written, not those for the individual runs.
plots		boolean	true	Write various plots for inclusion in .html output
appendtoresultsfile		boolean	false	If false, maxentResults.csv file is reinitialized before each run
maximumiterations	m	integer	500	Stop training after this many iterations of the optimization algorithm
convergencethreshold	c	double	1.0E-5	Stop training when the drop in log loss per iteration drops below this number
adjustsampleradius		integer	0	Add this number of pixels to the radius of white/purple dots for samples on pictures of predictions. Negative values reduce size of dots.
threads		integer	1	Number of processor threads to use. Matching this number to the number of cores on your computer speeds up some operations, especially variable jackknifing.
lq2lqptthreshold		integer	80	Number of samples at which product and

				threshold features start being used
l2lqthreshold		integer	10	Number of samples at which quadratic features start being used
hingethreshold		integer	15	Number of samples at which hinge features start being used
beta_threshold		double	-1.0	Regularization parameter to be applied to all threshold features; negative value enables automatic setting
beta_categorical		double	-1.0	Regularization parameter to be applied to all categorical features; negative value enables automatic setting
beta_lqp		double	-1.0	Regularization parameter to be applied to all linear, quadratic and product features; negative value enables automatic setting
beta_hinge		double	-1.0	Regularization parameter to be applied to all hinge features; negative value enables automatic setting
logfile		string	maxent.log	File name to be used for writing debugging information about a run in output directory
cache		boolean	true	Make a .mxe cached version of ascii files, for faster access
defaultprevalence		double	0.5	Default prevalence of the species: probability of presence at ordinary occurrence points. See Elith et al., Diversity and Distributions, 2011 for details.

applythresholdrule		string		Apply a threshold rule, generating a binary output grid in addition to the regular prediction grid. Use the full name of the threshold rule in Maxent's html output as the argument. For example, 'applyThresholdRule=Fixed cumulative value 1'.
togglelayertype	t	string		Toggle continuous/categorical for environmental layers whose names begin with this prefix (default: all continuous)
togglespeciesselected	E	string		Toggle selection of species whose names begin with this prefix (default: all selected)
togglelayerselected	N	string		Toggle selection of environmental layers whose names begin with this prefix (default: all selected)
verbose	v	boolean	false	Gived detailed diagnostics for debugging
allowpartialdata		boolean	false	During model training, allow use of samples that have nodata values for one or more environmental variables.
prefixes		boolean	true	When toggling samples or layers selected or layer types, allow toggle string to be a prefix rather than an exact match.
nodata	n	integer	-9999	Value to be interpreted as nodata values in SWD sample data