**INTRODUCTION TO GENERALIZED ADDITIVE MODELS**

**by Simon Moss**

|  |
| --- |
| **Introduction** |

**The need to include additive models**

Suppose you wanted to examine the motivation of research candidates over their candidature. In particular

* every week, candidates indicate their motivation on a scale of 1 to 10
* they also answer other questions that could affect motivation, such as their gender and IQ.

The following figure shows how the motivation of candidates tends to across time—but only in males with an IQ of 100. The data, obviously, do not conform to a simple pattern and, for example, deviate appreciably from the blue line. Consequently, many traditional statistical techniques, such as linear or multiple regression, are unlikely to be suitable.



Instead, perhaps you could use a set of equations, formulas, or models called polynomials. For example, the equation could be motivation = 1.3 + 0.5 x time + 0.04 x time2 + 0.002 x time3. The blue line in the following figure illustrates a polynomial. Again, however, the data seem to deviate too erratically across the horizontal axis to conform to a polynomial.



To characterize these unpredictable deviations and fluctuations across the horizontal axis, you might instead need to blend a series of curves. To illustrate, in the following figure, the three waves on the bottom, when summed, might create the line on the top. The waves on the bottom are called basic functions. The line on the top is called a spline.



As the following figure shows, a series of splines can be arranged in a sequence to characterize erratic patterns of data. In short

* intricate patterns of data can be represented as a sequence of fragments called splines
* each spline is a sum of many simpler patterns, called basic functions.
* because these basic functions are summed, these models are called additive
* the model can also include other predictors, such as gender and IQ, like a typical linear regression.



**The need to generalize these additive models**

 In the previous model, the outcome, motivation, was assumed to be a numerical variable, ranging from 1 to 10. But, sometimes, you might want to develop equations, formulas, and models to characterize other outcomes including

* categories, such as whether or not candidates complete their thesis
* ordered categories, such as Bachelor, Masters, and PhD degrees
* counts, such as number of degrees candidates have completed, and so forth.

That is, you want to generalize these additive models to many kinds of outcomes. The technique generalized additive models, sometimes called GAMs, was developed to achieve this goal. That is, generalized additive models are

* generalizable—that is, they can be applied to many kinds of outcomes, including numerical outcomes and categorical outcomes
* additive—that is, you can sum many basic functions to generate a sequence of splines.

|  |
| --- |
| **A simple example** |

**How to conduct a GAMs**

 To illustrate GAMs, this section presents an example using the software R. If you have not used this software before, perhaps read “Introduction to R”, available on the CDU webpage about “Choosing your research methodology and methods”. Then, in R, import your data, as shown below.



To conduct GAMs, enter the following commands

* library(mgcv)
* model1 = gam(motivation ~ sex + IQ + s(Years, bs = “tp”, k=10), data = Fake.data.3, family = gaussian, method = "REML", select = TRUE)
* summary(model1)
* plot(model1)

The following table explains each of these commands and subcommands.

|  |  |
| --- | --- |
| Command and subcommand | Explanation |
| library(mgcv) | * Loads or activates the mgcv package—a package that conducts gams
 |
| model1 | * A label the researcher concocts to name the model
* The benefit is the researcher can then refer to this model later—such as plot the model
 |
| gam | * gam is the command used to conduct gam, funnily enough
 |
| motivation | * the outcome or dependent variable
 |
| sex and IQ | * two linear predictors—variables that are linearly associated with the outcome
 |
| s(Years) | * calculates a smoothing function between one predictor—in this instance, Years—and the outcome
 |
| bs=”tp” | * bs is used to specify the basic function—the functions that generate each spline
* in this instance, “tp” means thin plate—but several other options are available
 |
| k=10 | * specifies the number of basic functions
* use a fairly high number; if the number is too high, GAMs will assign a coefficient of 0 to unnecessary basic functions
 |
| data = Fake.data.3 | * the name of the data file—in this instance, Fake.data.3
 |
| family = gaussian | * the probability distribution
* rather than gaussian, other options are common when the outcome is a category or count
 |
| method = "REML" | * the method used to estimate the parameters—and briefly discussed later
 |
| select = TRUE | * this subcommand simplifies the model
* in essence, if a term in the model, such as a predictor, does not improve the model, the coefficient is set to 0
* thus, this subcommand deletes unnecessary terms
* you can remove this subcommand
 |
| summary(model1) | * summarises the output of model 1, such as the main coefficients
 |
| plot(model1) | * plots the association between the smoothing function and the main predictor
* sometimes used to decide whether smoothing or GAMs is necessary
 |

Often, the outcome might vary erratically as a function of two predictors, such as time and distance from the equator. In this instance, both predictors would be included in the brackets after s, such as s(Years, Distance).

**Interpret the output**

 After you enter the summary command, R will produce some output. A sample of output appears in the following table.

|  |
| --- |
| Parametric coefficients: Estimate Std. Error t value Pr(>|t|)(Intercept) 3.25129 2.98269 1.090 0.292Sex **0.22161** 0.46122 0.480 **0.638**IQ **0.01142** 0.02984 0.383 0**.707**Approximate significance of smooth terms: edf Ref. df F p-values(Years) **1.37** 1.651 0.996 **0.464**R-sq.(adj) = -0.0662 Deviance explained = 12.3%-REML = 29.807 Scale est. = 0.94823 n = 20 |

 The most important output appears in bold. For example

* the Pr or p values exceed .05, indicating that neither sex nor IQ are related to motivation
* if these p values had been less than .05, the Estimates—the B coefficients—would indicate whether these relationships are positive or negative
* an edf—or estimated degrees of freedom—near 1 indicates the equation is not especially wiggly and, therefore, the smoothing function is unnecessary.
* similarly, a non-significant overall p value in the same table indicates the equation is not especially wiggly
* if so, another technique, such as linear mixed models, may be preferable to GAMs.

|  |
| --- |
| **Generalizing to other kinds of outcomes**  |

 In the previous example, the outcome was assumed to be a numerical variable. When the outcome is numerical, researchers often utilize a gaussian or normal distribution, corresponding to the code “family = gaussian”. However, when the outcome is not numerical, researchers might utilize another distribution. They might replace “family = gaussian” with “family = scat”, family = ocat”, “family = poisson”, and so forth. The following table clarifies when to utilize some of the distributions you can choose.

|  |  |  |
| --- | --- | --- |
| Distribution | Code | When to use this distribution |
| Gaussian or normal | family = gaussian | For numerical outcomes |
| Candidates t-distribution | family = scat | For numerical outcomes with many outliers—and thus thicker tails in the distribution |
| Poisson distribution | family = poisson | For count outcomes, such as number of university degrees. Smaller counts tend to be more frequent than larger counts |
|  Negative binomial distribution | family = negbin | For count outcomes, such as number of university degrees. But smaller counts may not be more frequent than larger counts |
| Tweedy distribution | family = tw | For count outcomes, such as number of university degrees. The data are clumped: A few low numbers and high numbers are appreciably more common than middle numbers for example |
| Ordered categorical distribution | family = ocat | For ordered categories as outcomes, such as highest degree can be ordered as Bachelor, Masters, PhD and so forth |
| Binomial | family = binomial | For categorical outcomes with two outcomes |
| Multinomial | family = multinom | For categorical outcomes in which the categories cannot be ordered from highest to lowest, like hair colour |
| Beta distribution | bamily = betar | For outcomes that vary from 0 to 1, like proportions.  |
| Cox proportional hazards | family = cox, ph  | For censored outcomes—when values higher or lower than a certain number are just equated to that number |

|  |
| --- |
| **Generalizing to other kinds of outcomes**  |

 The basic functions that generate the splines—and ultimately characterize the intricate patterns—can vary. These functions are not always simple waves. Indeed, these functions are not always only two dimensional, but can include three dimensional shapes or even more dimensions. Each kind of function can be used in specific circumstances. The following table outlines some of the options that are available.

|  |  |  |
| --- | --- | --- |
| Basic function | Code | When to use this function |
|  Thin plates | * bs = “tp”
 | * In many circumstances
* The default in this package
 |
| Adaptive smooths | * bs = “ad”

Here is an example* gam(y ~ s(x, bs = “ad”), data = fake.data)
 | * Useful when the degree to which the pattern is erratic or wiggly—called delta—varies across the horizontal axis
 |
| Random effects | * bs = “re”

Here is an example* gam(y ~ s(age\_group, bs = “re”)
 | * Useful when you want to include a categorical predictor—but represent only random effects of this predictor—a principle that would be meaningful only if you have learned about random effects or multi-level modelling
 |
| Factor smooth interaction | * bs = “fs”

Here is an example* gam( y ~ s(height, age\_grouup, bs = “fs”)
 | * Useful when you want to examine how the effect of some numerical predictor depends on some random categorical variable
 |
| **Spatial functions** |  |  |
| Markov random fields | * bs = “mrf”

here is an example—that also includes some other features, such as lists of neighbourhoods* gam(y ~ s(x, bs = “mrf”, xt = list(nb = nb)), data=data)
 | * Useful when you want to examine how some outcome, such as bacteria concentration, varies across regions like states
* Each state is discrete, but contiguous states should generate more similar outcomes than distant states
 |
| Soap films  | * bs = “so”, xt(bnd = my\_boundary)
 | * Useful when you want to explore how some outcome, such as concentration of bacteria, varies within some bounded shape—like a lake or river.
* Note that, if a river meanders, two segments of the river might be close in space but very different in bacteria
* These functions can model these circumstances well
 |
|  Spline on a sphere | * bs = “sos”
 | * Useful when you want to explore how some outcome, such as bacteria concentration, varies over a sphere, such as the Earth
 |
| Temporal functions  |   |   |
| Auto-correlation  | * bs = “gp”
 | * Represents auto-correlation across time—the observation that two scores are likely to be more similar if closer in time
 |
| Cyclic spines | * Bs = cc

Here is an example* gam(y~ s(time, bs = “gp”), s(month, bs = cc”)
 | * Useful when seasonal or cyclic effects are likely
 |

|  |
| --- |
| **Method=REML**  |

Often, when researchers conduct GAMs, they include the code “Method=REML”. REML is an acronym that means restricted maximum likelihood. You do not , at this stage, need to understand restricted maximum likelihood. But, you should appreciate the purpose of restricted maximum likelihood. To understand the purpose, consider the blue curve in the following figure. This curve is obviously not wiggly enough to correspond closely to the data. Consequently, researchers cannot utilize this curve to accurately predict motivation from duration in the course.



Now, consider the blue curve in the following figure. This blue curve is too wiggly—some of the wiggles might be too dependent on outliers or random events that affect motivation. For example, motivation might be low one day merely because a bowling bowl fell on the head of a candidate.



 Obviously, researchers need a balance—enough wiggles to explain the data accurately but not so many wiggles that perhaps the curve is too sensitive to random events. In strict terms, researchers strive to optimize a parameter called lambda or . This parameter represents the optimal extent to which the curve or equation should be wiggly. And REML attempts to ascertain this optimal level of .

**Tensor splines**

If your smoothing function comprises two predictors—such as s(Years, Distance)—GAMs assumes the association between each predictor and the outcome is equally wiggly. That is, GAMs uses the same for both Years and Distance. To override this default, you can replace the s with a te, referring to a tensor spline. That is, you would enter te(Years, Distance).

|  |
| --- |
| **Benefits of GAMs over alternatives**  |

**Benefits of GAMS over machine learning**

 Instead of GAMs, some researchers utilize a suite of approaches called machine learning—approaches that include neural networks, support vector machines, and random forests. These approaches have become fashionable in recent decades. Yet, as the following figure shows, relative to machine learning, GAMs offers some benefits and drawbacks



 Specifically, as this figure shows, some techniques, such as neural networks and other variants of machine learning, are very accurate: That is, they can predict outcomes, such as motivation of candidates very accurately

 Yet, these techniques are hard to interpret. That is, in most circumstances, the researcher cannot readily interpret the association between a specific predictor, such as IQ, and the outcome, such as motivation. They cannot readily answer questions like “To what extent will motivation improve if my IQ increases by one standard deviation?” or “If I am male, is IQ more or less important?” GAMs can resolve these questions to some extent, although not as effectively as traditional regression analysis.

|  |
| --- |
| **Other features**  |

In R, you can include other features as well. For example,

* to develop more meaningful output, you might need to utilize more complex plots and graphs in R
* bam is a command that is similar to gam, but quicker, although not quite as versatile yet
* another package in R can be used to generalize GAMs to nested effects and multilevel models
* a further package in R can be used to apply GAMs using a Bayesian approach

|  |
| --- |
| **Other references**  |

http://environmentalcomputing.net/intro-to-gams/