Variance models in `earth`

Stephen Milborrow

December 12, 2014

Contents

1 Introduction

A *variance model* models the variance of the response and can be used to estimate prediction intervals. The left plot of Figure 1 shows an `earth` fit with estimated prediction intervals. (We will discuss this plot in detail in Section 3.1.) The variance models in the `earth` implementation assume that the errors are independent but possibly heteroscedastic.

Use `earth`'s `varmod.method` argument to build a variance model. The variance model is kept with the `earth` model in its `varmod` field.

To build the variance model, `earth` models how the residuals vary with the predicted response. If we specify `varmod.method="lm"`, for example, `earth` first builds a MARS model as usual, then internally applies `lm` to the model's absolute residuals:

```
residual.model <- lm(abs(residuals) ~ predict(earth.model), ...)
```

The right plot of Figure 1 illustrates this residual model. The residual model allows us to estimate the average absolute value of the residuals at any predicted value, and thus the standard deviation.

Limitations of variance models There is more uncertainty in the variance model than in the main `earth` model. The right plot of Figure 1 is typical. It shows how noisy the residuals are. We expect the R-Squared of the `lm` regression here to be quite low. There is some uncertainty in the exact form of the residual model. Consequently there will also be some uncertainty in the estimated prediction intervals. Section 7.1 discusses techniques for checking the variance model.

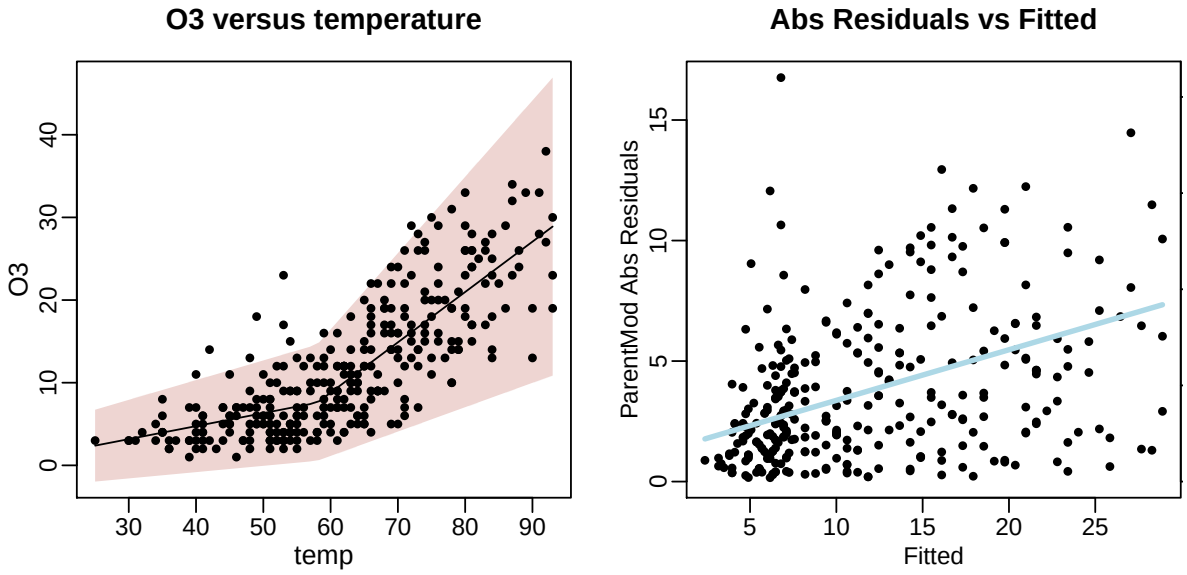

Figure 1: *Prediction intervals.*
Left: Ozone regressed on temperature with shaded 95% prediction intervals.
Right: Absolute residuals versus fitted values for the model in the left plot. The blue line is the linear residual model.

2 Theory and implementation

This section gives some details on the theory and implementation of `earth`'s variance models. If you want you can skip this for now and proceed directly to the example in the next section, Section 3.

For more on the theory of variance models, Carroll and Ruppert [1] and Davidian and Carroll [2] are recommended. Galecki and Burzykowski [4] is helpful for the likelihood theory.

In the `earth` implementation, since we are in a non-parametric setting, the methods in those references are extended to account for model variance (which is estimated with cross-validation). The methods presented here have currently been implemented only for `earth`, although they are in fact quite general.

2.1 Confidence intervals versus prediction intervals

There is an important distinction between the two types of intervals for predictions:

(i) intervals for the prediction of the mean response (called *confidence intervals*)

(ii) intervals for the prediction of a future value (called *prediction intervals*[1]).

To understand the distinction, suppose we have a model that predicts the selling price of homes in a given area based on the number of bedrooms, etc. (This example follows Section 3.5 of Faraway [3].)

(i) **Prediction of the mean response requires confidence intervals**. Suppose we ask the question "What would a house with given characteristics sell for *on average*?" The answer is the value predicted by the model. (Our interest is in `earth` models, but this of course applies to all models. For example, in a linear model $y = x^T\beta + \epsilon$, this prediction is $x_0^T\hat{\beta}$.) There will be some uncertainty in this predicted value because of model variance. The *confidence interval* of the prediction accounts for this model variance.

Model variance is a measure of how the model depends on the sample. With a different sample, the model will be slightly different. In a linear model there is uncertainty in our estimate of β. In an `earth` model it includes the uncertainty in the position of the knots, which variables were chosen for the model, etc.

(ii) **Prediction of a future value requires prediction intervals**. Because of noise, or irreducible error, the value of a *specific house* with the given characteristics will usually differ from the predicted value. (This noise is usually modeled as additive error ϵ. In a linear model for instance, this noise is represented by the random value of ϵ in $y = x^T\beta + \epsilon$.)

To figure out a *prediction interval* this noise must be added to the model variance. The prediction interval is bigger than the confidence interval.

[1] This terminology is not quite apt. It is what people use, but in some sense both are prediction intervals and both are confidence levels. Some call them wide and narrow intervals.

Another example is a model that estimates flood levels given the characteristics of the location (previous levels, nearby rivers, etc.). If we plan to build a flood wall, the confidence interval isn't enough — to be safe we need to use at least the upper 95% prediction interval.

2.2 Data structures

The various models are stored as follows:

```
earth model    Sometimes called the "parent model" in this context
   |
varmod         The variance model
   |
residmod       The residual submodel e.g. lm
```

The *residual submodel* is the regression of the absolute residuals on the fitted value. For example, it will be a `lm` model if `varmod.method="lm"`. Another example would be `varmod.method="gam"`, to adapt to non-linear changes in residual deviation.

The *variance model* is a wrapper for the residual submodel. It provides summary and plot methods, and takes care of rescaling absolute residuals to standard deviations (Section 2.5), and clamping to `min.sd` (Section 2.3).

The `summary.earth` function will display the variance model if present as part of the `earth` model.

You probably won't need to do this, but `summary(earth.mod$varmod)` will display the variance model directly (it invokes `summary.varmod`). And you can display the residual submodel directly with `summary(earth.mod$varmod$residmod)`. If `varmod.method="lm"`, the submodel is an `lm` model, and this call invokes `summary.lm`.

2.3 Minimum standard deviation

The standard deviation predicted by the variance models is forced to be at least a small positive value `min.sd`. This prevents negative or absurdly small estimated prediction intervals. The value of `min.sd` is determined when building the variance model as one tenth the average standard deviation:

```
min.sd <- 0.1 * mean(sd(residuals))
```

The `0.1` can be changed with `earth`'s `varmod.clamp` argument.

2.4 Iteratively Reweighted Least Squares

When building the residual model (as per the R code on page 3), `earth` uses Iteratively Reweighted Least Squares (IRLS). That is, it makes the call to `lm` repeatedly using the variance estimated from the previous call to determine the weights for the current call. Iteration stops when the `lm` coefficients change by less than 1%.

The iterations can be traced by specifying `trace=.3` in the call to `earth`, which will print something like this:

```
iter weight.ratio coefchange% (Intercept)      x
   1           10        0.00         1.7   0.17
   2           15       18.69         1.3   0.20
   3           17        3.90         1.3   0.21
   4           17        0.83         1.3   0.21
```

The `(Intercept)` and `x` columns show the estimated intercept and coefficients of the residual model (i.e., the results of the calls to `lm`). In this example, as is often the case, by the second iteration the estimates have settled close to their final values.

The `coefchange%` column shows the mean change in these values from the previous iteration. Iteration stops when the change gets below 1%. You can adjust the convergence criterion with `earth`'s `varmod.conv` argument, although you probably won't need to.

The `weight.ratio` column shows the ratio of the maximum to minimum weight. The weights are artificially clamped if necessary to prevent a few cases completely dominating.

Weighted least squares is necessary because in general the residuals of the residual model are themselves heteroscedastic. To determine the weights we need the variance of the residual model predictions. Happily, a characteristic of residual models is that this variance can be estimated from the predictions themselves — for regression on absolute residuals, this variance is proportional to the square of the predicted value (Carroll and Ruppert [1] Table 3.3).

No iteration is necessary when `varmod.method="const"`.

In the current implementation no iteration takes place with `varmod.method="earth"`. (This may change. We currently don't iterate the residual `earth` model because (i) the current implementation of `earth`'s `weights` is provisional, and (ii) under iteration the residual model tends to deteriorate to an intercept-only model, which isn't much use.)

2.5 Residuals

The `residuals` in the regression formula on page 3 include both irreducible error and model variance. The `earth` code forms the squared residuals as follows:

```
squared.residuals <- n / (n - df) * (y - meanfit)^2 + model.variance
```

The absolute residuals for the residual model are the square root of the `squared.residuals` (since `abs(residuals) == sqrt(residuals^2)`). Using squared residuals allows us to combine irreducible error and model variance by simple addition.

The degrees-of-freedom `df` is taken to be the number of MARS terms. This could be debated, but the precise `df` is not important with a reasonable number of observations `n`.

The `meanfit` of a prediction (at a fixed set of predictor values) is its average value in the hypothetical situation where we have a large number of datasets. We actually have

only one dataset at hand, so estimate it with repeated cross-validation: it is the mean of the out-of-fold predicted values.[1]

The `model.variance` for that prediction is its variance if we had a large number of datasets. It is estimated as the variance of the out-of-fold predicted values over `ncross` cross-validations.

The effect of hat matrix leverage is assumed to be ignorable.

There are other methods of forming the residuals for estimating prediction intervals. The above formula separates the contributions of the irreducible and model errors, analogously to the standard formula for the homoscedastic linear model $y = x^T\beta + \epsilon$. For that model, the estimated squared irreducible error $\hat{\sigma}^2 = \frac{n}{1-df}(y-\hat{y})^2$, the estimate model variance $var(x_0^T\hat{\beta}) = x_0^T(X^TX)^{-1}x_0\sigma^2$, and we sum these to get the estimated variance of predictions $\hat{\sigma}^2 + x_0^T(X^TX)^{-1}x_0\hat{\sigma}^2$.

2.6 Converting absolute residuals to prediction intervals

To estimate the prediction interval at a given prediction $\hat{y}$, we do the following:

(i) Estimate the mean absolute residual at $\hat{y}$ using the residual model described on page 3.

(ii) Assuming normality of the residuals, rescale the mean absolute residual to an estimated standard deviation with the formula

$$se(\hat{y}) = 1.2533\ mean(abs(residual(\hat{y})))$$

where the scaling factor $1.2533 = \sqrt{\pi/2}$ is the ratio of the standard deviation to the mean absolute deviation for normal data (Geary [5]). The factor can also be estimated with the R expression

```
1 / mean(abs(rnorm(1e8)))   # evaluates to 1.2533
```

(iii) Convert the standard deviation to an estimated prediction interval for a given level α:

$$interval(\hat{y}) = \hat{y} \pm z_{\alpha/2}\ se(\hat{y}).$$

The above steps take place in `predict.varmod`, which is invoked by `predict.earth` when its `interval` argument is used.

2.7 Why regress on the absolute residuals?

Since our aim is to estimate variance (or standard deviation), why don't we regress directly on the squared residuals? That seems like the right way to go, since the expectation of the square of the residuals is their variance, up to a degrees-of-freedom correction, and is the approach suggested by some authors.

[1] An *out-of-fold prediction* is made from an observation not in the set used to build the model. In `ncross` cross-validations, there will be a total of `ncross` out-of-fold predictions for each observation.

In practice we have found that regressing on the squared residuals (or log residuals as suggested by some) tends to give results worse than absolute residuals. The chi-squared distribution of the squared residuals is too far from gaussian. An outlying residual when squared becomes even more outlying, affecting robustness of the residual model. The absolute residuals are much closer to normality.

The cube root of the squared residuals is even closer (Wilson and Hilferty [6]). However the `earth` implementation sticks with absolute residuals because they are close enough in practice, and slightly more intuitive.

For the `earth` variance model, a regression based approach (rather than a likelihood approach) was chosen for its conceptual simplicity, and its flexibility given the ease with which we can plug in different R regression functions. Also, likelihood estimation in this setting is less robust because it is sensitive to departures from the assumed distribution (Carroll and Ruppert [1] Section 2.4).

2.8 Miscellaneous issues

TODO This section still needs to be written.

Explain what a prediction interval is.

The variance model estimates average not expected variance. All variability measured by the residual model is measured on the fold models, not the final model on the full training set (Section 2.5).

For valid results, the red and blue lines in the `plot.earth` residuals plot should be approximately on the center line.

The bias problem. Like all non-parametric models, MARS balances bias and variance — not too much variance; not too much bias. Therefore the model will be biased, at least a bit. With MARS models we especially see this at sharp corners in the response. Bias affects the estimation of conditional variance, a fundamental problem (Wasserman "All of non-parametric statistics" Section 5.7).

3 A variance model example

The code below builds a simple univariate model to estimate the ozone level `O3` from the temperature `temp`. We will use this model as a running example through this chapter.

```
data(ozone1)
set.seed(1) # optional, for cross-validation reproducibility
earth.mod <- earth(O3~temp, data=ozone1, nfold=10, ncross=30, varmod.method="lm")
```

Note the `varmod.method` argument in the call to `earth`. The `nfold` and `ncross` arguments are also necessary when building a variance model (because repeated cross-validation is used to estimate model variance). The `ncross` argument should be at least 30 in this context, based on the rule-of-thumb that we need 30 measurements to adequately estimate variance.

We use `varmod.method="lm"` above with the assumption that the standard deviation of O3 increases linearly with temperature. (Actually, the relationship is probably more complex, but, as can be seen in the right plot of Figure 1, the residuals are noisy, and determining the exact nature of the relationship may not be possible from the data alone. A linear regression is a good first choice.)

The left plot of Figure 1 was produced by the following call to `plotmo`. Note the `level=.95` argument, which tells `plotmo` to show the 95% prediction intervals.

```
plotmo(earth.mod, col.response=1, level=.95)
```

3.1 Comments on the example

In the left plot of Figure 1, we see some outlying points in the 40 to 60 degree temperature range.

On the far left of the plot the estimated lower prediction limit is below zero, which is impossible (the ozone level cannot be negative). So we have to be sensible about how we interpret the prediction intervals.

Residual deviation seems somewhat overestimated at low and high temperatures. This may be a consequence of our decision to use `lm` for the residual model. A GAM model might have been a better choice (`varmod.method="gam"`), although less easily interpretable and more prone to overfitting.

The residual model assumes that the residuals are symmetric. We perhaps see the effects of that assumption for temperatures in the 70 to 80 degree range, where the points are dispersed asymmetrically about the regression line — the estimated prediction band is a little too big for points below the line, and a little too small for points above the line.

The assumption of symmetry follows from the fact that the residual model regresses on the absolute values of the residuals, thus making no distinction between positive and negative residuals. Of course the implementation could be modified to generate one-sided intervals with separate regressions on the positive and negative residuals. That may be an option worth adding to the `earth` code. A disadvantage is that it halves the

data for regression, in an already noisy situation. Also, the skewed distribution of the one-sided residuals may adversely affect the regression.

The shape of the main `earth` model hinge suggests that instead of `earth` here for the main model we could use a standard linear regression on say the square or cube root of `temp`. But we ignore that for our current purposes.

This example is univariate (one predictor). With multivariate models (multiple predictors), displaying the prediction intervals isn't so easy (`plotmo` will do it, but the results can be confusing). For such models, `plot.earth`'s residual plot is helpful for checking the variance model, as will be discussed in Section 5.2.

3.2 Extensions to `summary.earth` for variance models

A call to `summary(earth.mod)` shows the variance model for our example:

```
> summary(earth.mod)

    ... usual stuff not shown ...

    varmod: method "lm"    min.sd 0.469    iter.rsq 0.138

    stddev of predictions:
                coefficients iter.stderr iter.stderr%
    (Intercept)        1.590      0.3454           22
    O3                 0.263      0.0364           14

                              mean   smallest   largest   ratio
    95% prediction interval   18.4        8.7        36    4.14

                                           68%    80%    90%    95%
    response values in prediction interval   70%    82%    92%    97%
```

The various items in the above summary are described below.

(i) The `iter.rsq` figure is the R-Squared of the weighted `lm` variance model. The low value of `0.138` isn't unusual here because of the difficulty of building a model from the main model residuals, which tend to be noisy (right plot of Figure 1).

The prefix `iter.` reminds us that this R-Squared is for the final iteration of IRLS (Section 2.4). As always with IRLS there is some concern about the validity of this R-Squared value. This is because the uncertainty in the iterated estimation of the regression parameters isn't fully accounted for in the calculation of the final R-Squared.

In general, inference on residual models is a difficult problem, whether one uses regression based methods, as we do here, or likelihood based methods (e.g. Galecki and Burzykowski [4] Section 7.8).

(ii) The standard deviation estimated by the variance model is never allowed to be less than `min.sd`, in this case `0.469`. (Section 2.3 describes `min.sd`.)

(iii) The `stddev of predictions` table gives the standard deviation of the predictions made by the `earth` model. In this example, it says the standard deviation of the predicted `O3` level is estimated to be `1.590 + 0.263 * O3` in units of `O3` concentration. This is the core of the variance model.

The `iter.stderr` shows the standard error of the coefficients, much like the output of `summary.lm`. These standard errors are calculated from the final model of the IRLS iteration. As in `iter.rsq` above, there is some concern about the validity of these numbers — they may be too small.

The `iter.stderr%` column shows `iter.stderr` as a percentage of the coefficient value. For instance, in the first line of the table:

```
iter.stderr%  =  iter.stderr / coefficient  =  0.3454 / 1.590  =  22%
```

(iv) The `95% prediction interval` table shows the `mean`, `smallest`, and `largest` estimated 95% prediction intervals for the `O3` response. In the left plot of Figure 1, the smallest prediction interval is at the extreme left and the largest is at the extreme right, since estimated variance increases with the response.

The ratio of the `largest` to the `smallest` is also shown. This is a measure of overall heteroscedasity. The current figure `4.14` indicates considerable heteroscedasity.

(v) The `response values in prediction interval` table is a sanity check of the variance model. (The table looks ok in this example.) It shows what fraction of the training response values (`O3`) are in some standard prediction intervals. The 95% interval corresponds to the bands in the left plot of Figure 1 because `level=.95` was used to generate that plot.

To generate the intervals for this table, `predict.varmod(type="pint")` is used (not `type="training.pint"`), so this table is also a check of interval bias (Section 2.8).

We can print the table for new data by invoking `summary.earth` with a `newdata` argument. This will also print R-Squared on the new data, as shown in the following example. (The example uses bogus new data which is just a subset of the training data.)

```
> summary(earth.mod, newdata=ozone1[sample.int(nrow(ozone1), 100), ])

RSq 0.731 on newdata (100 cases)

                                    68%    80%    90%    95%
newdata in prediction interval      76%    84%    91%    95%
```

4 Prediction intervals and `predict.earth`

Use `predict.earth`'s `interval="pint"` and `level` arguments to get estimated prediction intervals on new data. For example (using the model built on on page 9):

```
predict(earth.mod, newdata=ozone1[1:100,], interval="pint", level=.95)
```

Here `predict` returns a dataframe with three columns showing the fit, and the lower and upper prediction limits for the given `level = .95`:

```
   fit    lwr  upr
1 4.75 -0.819 10.31
2 5.53 -0.438 11.50
3 6.95  0.247 13.65
        ...
```

The `predict.earth` function calls `predict.varmod` internally with the given `interval` and `level` arguments. See `help(predict.varmod)` for details.

4.1 Plotting the prediction intervals

We can plot the prediction intervals with `plotmo` (left plot of Figure 1):

```
plotmo(earth.mod, col.response=1, level=.95)
```

We can also plot the intervals manually (Figure 2):

```
predict <- predict(earth.mod, newdata=ozone1, interval="pint", level=.95)
# x values have to be ordered to plot lines correctly
order <- order(ozone1$temp)
temp <- ozone1$temp[order]
O3 <- ozone1$O3[order]
predict <- predict[order,]
in.interval <- O3 >= predict$lwr & O3 <= predict$upr
plot(temp, O3, pch=20, col=ifelse(in.interval, "black", "red"),
  main=sprintf(
  "Prediction intervals\n%.0f%% of the training points are in the estimated 95%% band",
  100 * sum(in.interval) / length(O3)))
lines(temp, predict$fit)          # regression curve
lines(temp, predict$lwr, lty=2)   # lower prediction intervals
lines(temp, predict$upr, lty=2)   # upper prediction intervals
```

4.2 The `interval` argument

The `interval` argument instructs `predict.earth` to return prediction intervals. This argument gets passed to `predict.varmod` as its `type` argument.

Exactly what intervals are returned is a bit confusing. This chart may help:

interval	center	stddev
"pint"	prediction	to.sd(predicted.abs.resid, newdata)
"training.pint"	meanfit	to.sd(predicted.abs.resid, traindata)
"training.cint"	meanfit	sqrt(model.var)

The `center` column shows the center of the interval:

- For `"pint"`, the center is the value predicted by the `earth` model
- For `"training.pint"` and `"training.cint"`, the center is the mean predicted value on the out-of-fold cases during cross-validation while building the model. (Section 2.5).

The `stddev` column shows the standard deviation of the residuals used to generate the prediction or confidence interval, Section 2.6 (iii).

This standard deviation is estimated as shown in the table, where

- `predicted.abs.resid` is the mean absolute residual predicted by the residual model
- `to.sd` is the function that scales the estimated mean absolute residual to a standard deviation, Section 2.6 (ii).

The `plotmo` function uses `predict.earth` with `interval="pint"`. The `plotmo` prediction intervals will thus be misleading unless the blue and red lines in `plot.earth`'s residual plot are more-or-less along the axis.

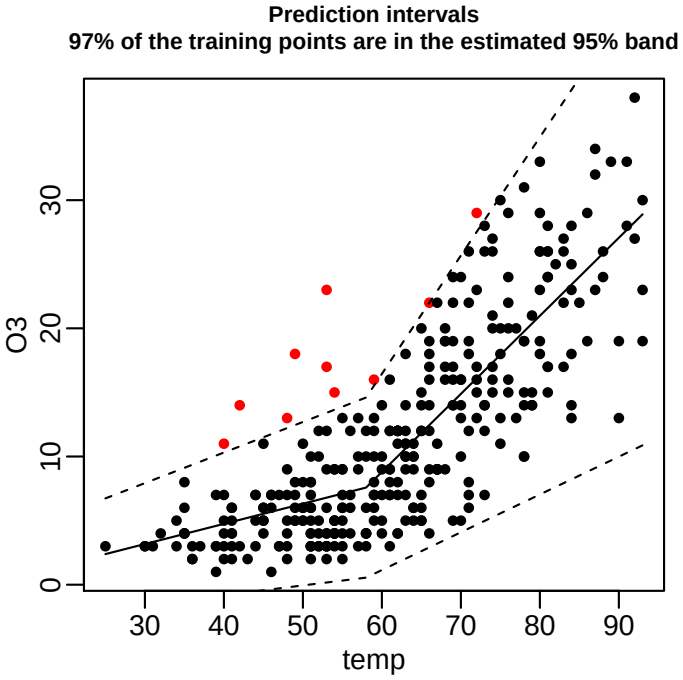

Figure 2: *Prediction intervals using* `predict.earth`.

The plot was produced with the code on this page.

It is essentially the same as the left plot of Figure 1, which was produced with `plotmo`.

5 Plotting variance models

This section discusses some of the plots that can be used with variance models. They are important for establishing credibility (or otherwise) of the variance model.

5.1 Extensions to `plotmo` for variance models

As we have already seen, `plotmo` will draw prediction intervals if given the `level` argument (pages 9 and 12).

`Plotmo` also knows how to draw prediction intervals for some other kinds of model, not just `earth` models. See its vignette for details.

5.2 Extensions to `plot.earth` for variance models

Use the `level` argument of `plot.earth` to display prediction bands in the residuals plot (left plot of Figure 3):

```
plot(earth.mod, which=3, level=.95)    # which=3 selects just the residual plot
```

In this plot:

- The darker grayish band shows the confidence limits; the wider pink band shows the prediction limits (Section 2.1 "Confidence levels versus prediction levels"). In this example the confidence limits widen at extreme low and high fitted values, indicating greater model uncertainty at those values.

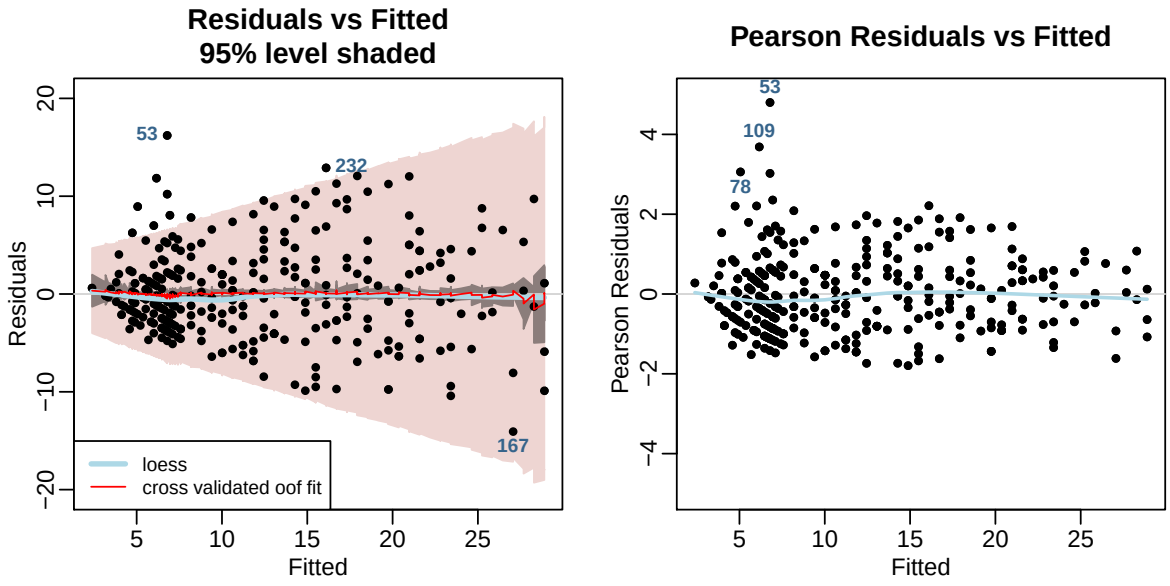

Figure 3: `plot.earth` *residuals with a variance model*
Left: `plot(earth.mod, which=3, level=.95)`
Right: `plot(earth.mod, which=3, pearson=TRUE)`

- The blue line is a lowess fit to the residuals. In this example, the line remains close to the center line, so indicates that the model is a good fit. (Remember that since this is a residual plot, the center line corresponds to the predicted fit.)
- The red line shows the cross-validation `meanfit` (Section!2.5). Once again, the line is close to the center line, indicating that the mean out-of-fold predictions generated by the fold models approximately match those of the final model on all the data.

5.3 The `pearson` argument of `plot.earth`

Use the `pearson` argument to divide each residual by its estimated standard deviation (right plot of Figure 3):

```
plot(earth.mod, which=3, pearson=TRUE)
```

If the variance model is correct, the pearson residuals will be homoscedastic (up to a hat matrix correction for leverage, which is usually minor). In this example the prediction bands are not displayed to make it easier to visually detect heteroscedasity, although for your purposes you may want to add a `level` argument to the call to `plotmo`.

We mention that there is inconsistency in the literature on the meaning of the terms "Pearson residuals" and "Studentized residuals".

5.4 The `info` argument of `plot.earth`

Figure 4 is the same as the right plot of Figure 3 but also includes the argument `info=TRUE`:

```
plot(earth.mod, which=3, pearson=TRUE, info=TRUE)
```

This tells `plot.earth` to display additional information:

(i) The bottom of the plot shows the distribution of fitted values. Most are bunched near the left of the graph. We realize that the outlying points in this region are less important than they may seem at first — because of the high number of points in the region, the outliers represent only a small fraction of the points.

(ii) Also shown is Spearman Rank Correlation of absolute residuals with the fitted values. This is a measure of heteroscedasity: the correlation will be positive if the residuals tend to increase as the fitted values increase. Similarly, a negative correlation would indicate decreasing variance (much less common). Remember that this correlation is subject to sampling variation.

In the current graph, the displayed value `0.05` is small. It indicates virtually no heteroscedasity of the residuals after pearson normalization.

If we used `info=TRUE` on the raw residuals in the left plot of Figure 3 (not shown), the displayed Spearman correlation would be `0.38`, confirming that there is correlation between the absolute residuals and the fitted values, i.e., the raw residuals are heteroscedastic.

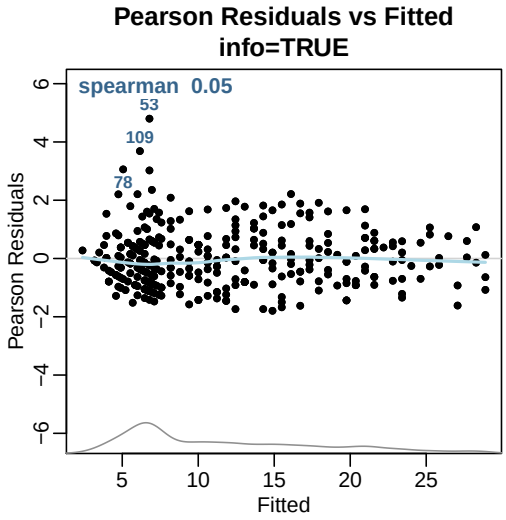

Figure 4: *The* `info` *argument of* `plot.earth`*.*

Same as the right plot of Figure 3 but includes an `info=TRUE` *argument.*

The density is plotted along the bottom.

Also shown is the Spearman Rank Correlation of absolute residuals with fitted values.

The Spearman correlation is invariant to monotone transforms to the response, unlike the more conventional Pearson Correlation Coefficient. Thus it won't change if measured on the squared or log residuals, for example.

The Spearman correlation measures only monotone variance trends (it won't detect variance that increases and then decreases by the same amount), so ultimately your eyeball is the best detector of heteroscedasticity, although it can be deceived by varying degrees of density along the horizontal axis.

(iii) The linear regression line is drawn if the plot shows absolute residuals (`which=5` or `8`, see the next section and Figure 5).

5.5 Plotting absolute residuals

The `plot.earth` function can also plot absolute residuals (Figure 5):

```
plot(earth.mod, which=5, info=TRUE}   # which=5 for absolute residuals
```

See the description of the `which` argument on the `plot.earth` help page for further possibilities.

With `info=TRUE`, a robust linear regression line is also plotted and its slope displayed. The absolute residuals are regressed against the fitted values with robust linear regression to show the overall trend unaffected by outliers. A standard (non-robust) linear fit would be steeper.

The lowess curve in this example indicates that variance decreases at low and high fitted values relative to the robust linear fit. But that conclusions assumes that the lowess curve is not overfitting, which is hard to tell.

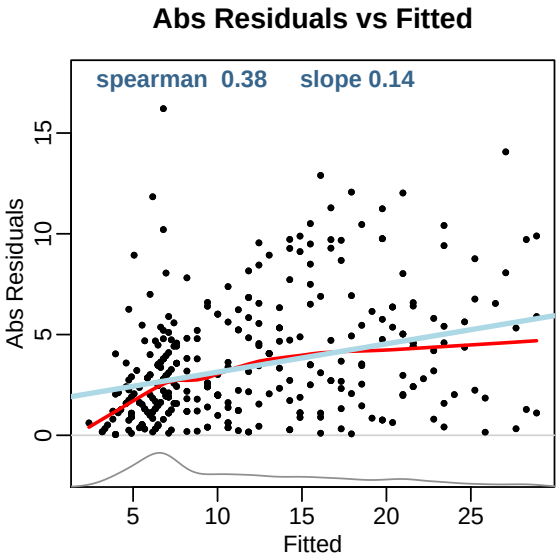

Figure 5: *A plot of the absolute residuals* `plot.earth(..., which=5)` *with a lowess line (red line).*

The `info=TRUE` *argument was also specified, so we see some additional information, including a robust linear regression line (blue line).*

The lines show that variance increases with fitted values.

5.6 The `plot.varmod` function

Use `plot.varmod` to display the variance model embedded in the `earth` model. For example (Figure 6):

```
plot(earth.mod$varmod)    # invokes plot.varmod
```

In the top left plot of Figure 6, the variance model is a straight line, because we used `varmod.method="lm"` when we called `earth`. (The term *parent model* in these plots refers to the main `earth` model.)

The axis on the right of the plot shows the standard deviation. (Section 2.6 explains how the absolute residuals are rescaled to standard deviation.)

This plot is similar to Figure 5, except that Figure 5 shows a *robust* linear regression line.

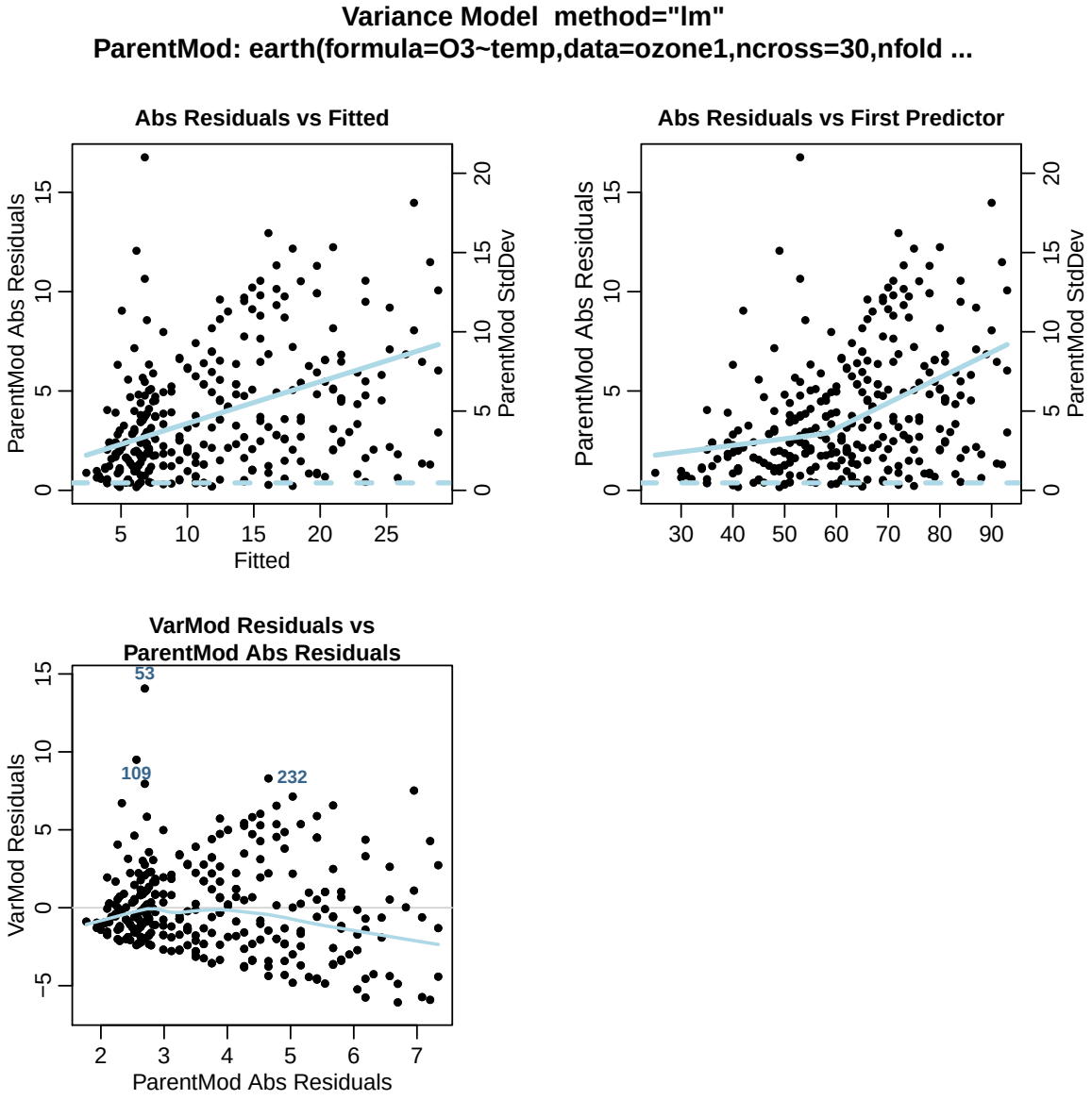

Figure 6: `plot.varmod`

The horizontal dashed blue line shows the clamping level set by `min.sd` (Section 2.3). But in this example is so happens that clamping of predicted standard deviations is unnecessary because the solid blue regression line stays well clear of the dashed blue line.

The top right plot shows how variance changes as the first predictor changes. In this example, the first and only predictor is the temperature `temp`. For multivariate models, similar plots can be generated for all predictors as follows. The `type` argument gets passed to `predict.varmod`.

```
plotmo(earth.mod$varmod, col.response=2, type="abs.residual"))
```

The bottom left graph shows the residuals of the variance model (the residuals of the `lm` regression on the parent `earth` model residuals). The blue line is a lowess fit. It curves here in the same way as it curves in Figure 5.

If `info=TRUE` is passed to `plot.varmod`, extra information is displayed, including lowess fits in the first two plots.

5.7 Multimapped variances

Figure 7 shows simulated data where the response has a non-monotonic relationship with the predictor. The same value of the response is associated with more than one value of variance — there is a one-to-many, or *multimapped*, relationship between the response and the variance. Non-monotonicity of the response is not necessary for multimapped variances, although it makes them more likely.

A real-world example: the average rainfall at a certain location is the same for the months of March and November, but the amount of rain from year to year varies more in March than in November.

The standard residual plot for the model (left side of Figure 8) is not so helpful in exposing the pattern of heteroscedasity. The right plot is much more informative. Here

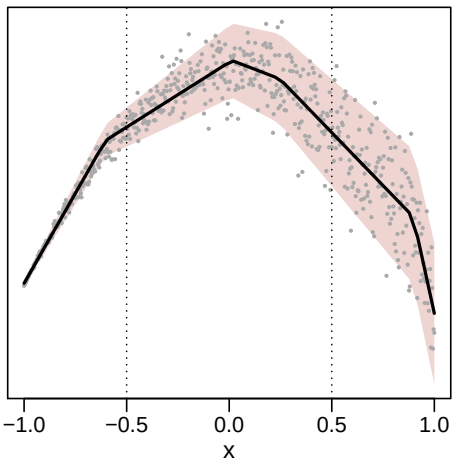

Figure 7: *Data with multimapped variances. The black line is an* `earth` *fit.*

The variance is quite different, for instance, at `x` *equal to* `-0.5` *and* `0.5`*, although the mean response is about the same.*

The variance must be modeled as a function of `x`*. We used* `varmod.method="x.lm"` *(Section 6.1).*

It can't be modeled as a function of the mean response (don't use `varmod.method="lm"`*).*

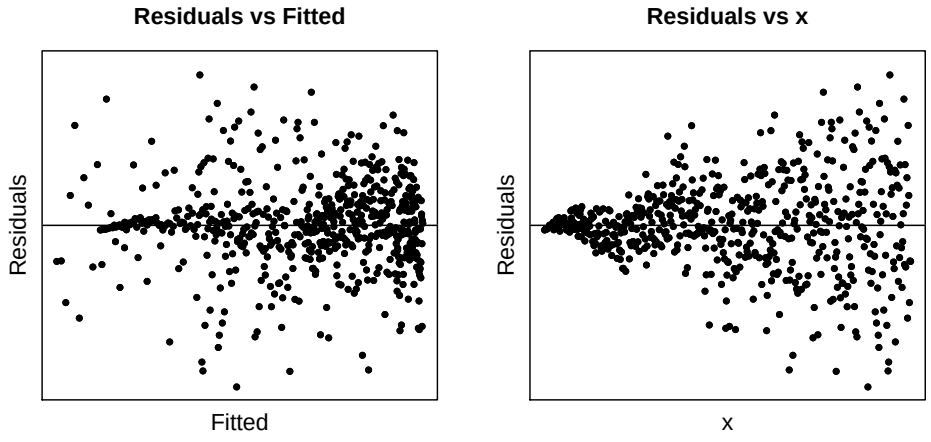

Figure 8: *Residual plots for the multimapped data in the above figure.*
Left: Residuals versus the fitted values. Not so informative.
Left: Residuals versus the predictor. Better.

the residuals are plotted against the predictor, instead of the fitted values. We clearly see that variance increases with the predictor.

The two plots in Figure 8 were generated with the following lines of code. The second line uses `plot.earth`'s `versus` argument to tell it to plot the residuals against the predictors (in this example there is only one predictor).

```
plot(earth.mmap, which=3)            # default    residuals versus predictions
plot(earth.mmap, which=3, versus="") # versus=""  residuals versus predictors
```

With `versus="*"`, the residuals are plotted against the MARS basis functions. See the `earth` help page for details.

With multivariate models (multiple predictors), determining if multimapping is occuring can be difficult. In the ozone data for instance, the O3 variance increases and then decreases as the day-of-the-year `doy` increases (Figure 9). However, with the other variables included in the model, the multimapping effect of the day-of-the year is probably negligible compared to the effect on variance of the other variables. Thus we can use `varmod.method="lm"`, not `varmod.method="x.lm"` (Section 6.1). TODO Verify that this is indeed the case.

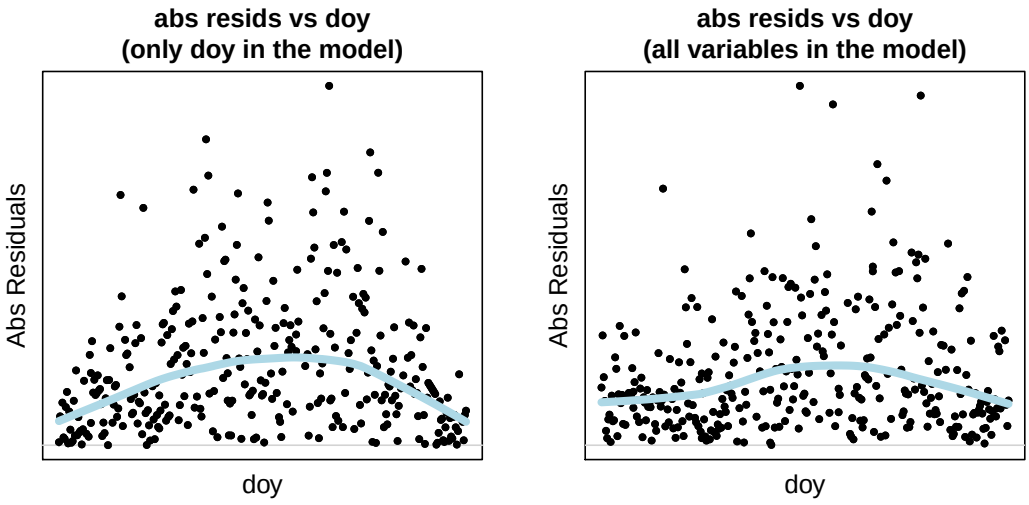

Figure 9: *Absolute residuals of ozone models. The blue lowess line shows that variance increases then decreases as the day-of-year increases.*

Left: *Just* `doy` *in the model:* `O3~doy`

Right: *All variables in the model:* `O3~.`
The overall variance is now less dependent on `doy`.

6 Arguments for the variance model

This section discusses some aspects of `varmod.method` and other arguments for the variance model.

For simplicity, our examples thus far have used `varmod.method="lm"`, There are other possibilities, see the `earth` help page. You may need to experiment with different options on your data.

6.1 Variance as a function of x or of $\hat{y}$?

When possible you should model variance as a function of the fitted response rather than of the predictors. That is, use the non-`x.` varmod methods (for example, prefer `varmod.method="lm"` to `"x.lm"`). This allows the main `earth` model to do the work of estimating the response from the predictors, leaving the variance model to do the generally simpler job of estimating the variance from the fitted response.

However, the `x.` varmod methods must be used for multimapped variances, where the residual variance has a many-to-one relationship with the response and thus cannot be modeled as a function of the response (Section 5.7). This will often be the case when the response is non-monotonic (e.g. it increases then decreases as a predictor increases).

6.2 `varmod.method="rlm"`

The `varmod.method="rlm"` variance model is like `varmod.method="lm"`, but uses robust instead of standard linear regression on the absolute residuals. The code uses `rlm` in the `MASS` package.

The `rlm` estimated prediction intervals will tend to be narrower than the `lm` intervals. This is because a standard linear regression line will be pulled by outlying residuals, whereas a robust line will tend to follow the general pattern.

It is difficult to say which method is better. The robust approach may at first seem better: the general pattern of residual variation is what we are interested in. However, the outliers might be the very residuals that matter, and the prediction intervals from the robust model may be optimistic.

6.3 `varmod.method="power"`

In many datasets with a positive response, standard deviation increases as a power of the mean response, at least approximately:

```
residual.std.dev  <-  intercept + coef * response ^ exponent
```

where the parameters `intercept`, `coef`, and `exponent` depend on the distribution of the data. This is a *power-of-the-mean* residual model (more pedantically, a power-of-the-mean model with an offset). In a Poisson distribution the standard deviation increases

with the square root of the response (`exponent = .5`). In a Gamma or lognormal distribution, the standard deviation increases linearly with the response (`exponent = 1`).

Often when applying `earth`, we don't know the exact distribution of the data but nevertheless can model the residuals accurately enough using a power-of-the-mean model. Use `varmod.method="power"` to estimate the `exponent` and other parameters. Internally, `earth` will make the estimates using a non-linear regression on the absolute residuals by means of `nls` in the standard `stats` package.

We illustrate with simulated data:

```
set.seed(1) # optional, for reproducibility
x <- 1:300
y <- x + (10 + 10 * sqrt(x)) * rnorm(length(x))   # y equals x + noise
earth.power <- earth(y~x, linpreds=T, nfold=10, ncross=30, varmod.method="power")
```

Note that when generating the data in the code above we (somewhat arbitrarily) used `intercept = 10`, `coef = 10`, and `exponent = 0.5` (square root). A call to `summary(earth.power)` yields

```
varmod: method "power"

stddev of predictions:
             coefficients iter.stderr iter.stderr%
(Intercept)        10.636       17.16          161
coef                5.261       5.361          102
exponent            0.619       0.173           28
```

The estimated parameters differ somewhat from those used to generate the data. For example, the estimated exponent is `0.619` rather than `0.5`. Also, even though we have a decent sized dataset (300 cases), we have large standard errors.

The example illustrates that estimating these parameters accurately isn't possible from typically noisy residuals. There is simply not enough information in the residuals to pinpoint the underlying distribution. So in general we can't expect too much of `varmod.method="power"`, although it may give us some understanding of the distribution. Another problem is that the internal call to `nls` on noisy residual data sometimes fails to converge, or causes the message `Error in numericDeriv`.

Also provided is `varmod.method="power0"`. This is the same as `"power"` but without the intercept term, to force a zero offset.

Use the special value `trace=.31` to trace the `nls` iterations while estimating the power model. This will also cause plotting of weights.

The power-of-the-mean model is for data with a positive response. When estimating the model, the `earth` function forces negative predicted responses to zero, because in general one can't take the power of a negative number. It will issue an error message if more than 20% of the responses are negative. This allows for some model error that causes a few negative predictions that in theory are always non-negative.

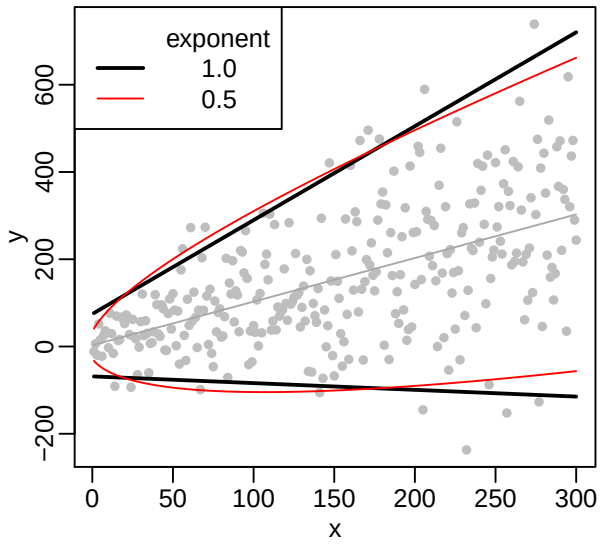

Figure 10: *Prediction intervals with two different settings of* `varmod.exponent`

Large changes to the exponent typically cause relatively small changes to the estimated prediction intervals.

The data here were generated by the code on page 23. The true value of the exponent is 0.5.

6.4 `varmod.exponent`

If the power-of-the-mean model described in the previous section applies to the data and the exponent is known, we can use `varmod.method="lm"` and specify the exponent with `earth`'s `varmod.exponent` argument. Earth applies the specified exponent to the right side of the formula when building the residual model.

For example, if you expect the standard deviation to increase with the square root of the response, use `varmod.method="lm"` and `varmod.exponent=.5`. (Negative predicted values will be treated as 0, and you will get an error message if more than 20% of them are negative.)

We can get an estimate of the exponent by generating a preliminary model using `varmod.method="power"`. We may want to round the estimate from the preliminary model. For example, an estimated exponent of `0.6`, could indicate that the standard deviation increases with the square root of the response (the difference between `0.6` and `0.5` being due to sampling variance). So we would build our `earth` model using `varmod.method="lm"` and `varmod.exponent=.5`. We wouldn't go so far as to infer that the data has a known distributional form — all we are hoping to do is solve the practical problem of estimating prediction intervals.

Exact specification of the exponent is usually not important. Changes to the exponent usually make relatively small changes to the estimated prediction intervals (Figure 10).

7 Assumptions and checking the model

When building a variance model, the methods described in this chapter make the following assumptions. Most of these assumptions apply in any case to `earth` models, but become more critical when building variance models.

(i) The errors are independent. Time series would be a no no.

(ii) The errors are symmetric. Dots should be dispersed roughly symmetrically about the center line in the residuals plot. The residual model uses absolute residuals, so makes no distinction between positive and negative residuals.

(iii) Error variance at a given mean response is approximately gaussian. One instance where this assumption is used is when converting the absolute residual predicted by the residual model to a standard deviation (Section 2.6).

(iv) The predicted value is close to the true mean value — if our main `earth` model is no good, there's no hope for a variance model. This assumption is necessary for example when using `predict.earth`'s `interval="pint"` argument.

(v) Cross-validation gives a reasonable estimate of model variance. This isn't too important if the model error is much smaller than the irreducible error, as it is in our running example. Use the residuals plot of `plot.earth` to check this. We mention that our experience on simulated data has been that cross-validation often underestimates model variance.

(vi) There is enough information in the residuals to form a decent residual model. The residuals are usually noisy, and the residual model may have a low R-Squared, but we assume it is still usable.

(vii) The residual model doesn't overfit. Overfitting is unlikely with `varmod.method="lm"`, although a possibility with other methods. Use plots to check that the residual model has no implausible curves or kinks.

7.1 Checking the variance model

TODO Consider merging this section with the above section.

The data may suffice to build an adequate main `earth` model but not be sufficient to build a valid variance model. This is especially true with smaller samples. Earth with the `varmod` argument will quite happily build a variance model, but we need to check the validity of that model.

For example, in the right plot of Figure 1, is there really enough information in the residuals to build a valid residual model? Probably so, but is there enough information for a non-linear residual model that curves to fit the residuals? Maybe also so, but we need to verify the model.

The first thing to do is to check `earth`'s residual plot:

(i) The blue lowess should be approximately straight and on the axis, indicating that the model fits the data. If the line is curved, the estimated prediction intervals won't be trustworthy.

(ii) The red cross-validation line should be approximately on the axis. This indicates that the out-of-fold predictions generated by the cross-validation models approximately match those of the final model. In left plot of Figure 3 we see a little unsteadiness at the far right of this line (this is caused by model variance: the cross-validation models vary a bit in this area of the curve). In this example, the unsteadiness is small enough not to worry about.

(iii) In the residuals plot, the prediction bands should match the general pattern of the point cloud.

We can check interval coverage by looking at the chart printed by `summary.earth`. For example:

```
                                        68%    80%    90%    95%
Response values in prediction interval  70%    83%    88%<   97%
```

The "<" printed above by `summary.earth` points out that only 88% of the training data is covered by the 90% prediction interval. The estimated prediction interval is too small. This is a hint that there may be some overfitting in the variance model, although some small variation like this is not unexpected and not really an issue.

Remember that this chart is for the training data, and so is only a sanity check for what would happen with new data. If possible, we should also print and check the table with new data. Do this by passing a `newdata` argument to `summary.varmod` (Section 3.2). This is a highly recommended check of the plausibility of the variance model.

8 Miscellaneous

8.1 Linear models with heteroscedasity

The standard `lm` function doesn't support variance models for heteroscedastic data. (It does support weights, and thus with manual IRLS you can estimate residual variance, but only point-wise for observations in the training set.)

Instead we can use `earth`'s `linpreds=TRUE` to argument to build a linear model with `earth`.[1] Together with `varmod` argument, this will build a variance model, allowing us to get prediction intervals on new data for linear models.

The `gls` function in the `nlme` package should also be considered.

8.2 Heteroscedasity when building the `earth` model

Although `earth`'s variance model estimates heteroscedasity, it doesn't actually account for it when building the MARS model. Although there is loss of efficiency, heteroscedasity generally does not affect estimation of the model too much. (Your mileage may differ.) It does affect inference, which we don't really do in `earth` anyway. Weights are not yet fully implemented in `earth`, so IRLS with `earth` isn't yet possible.

[1] There is a chapter on this in the vignette "Notes on the earth package".

Acknowledgments

Thanks to Glen Eanes and Kyra Stull for the prototype homoscedastic model and for helpful discussions. Thanks to Glenda Matthews for her suggestions. And thanks to Trevor Hastie for his feedback.

References

[1] Raymond J. Carroll and David Ruppert. *Transformation and Weighting in Regression*. CRC, 1988. Cited on pages 4, 6, and 8.

[2] M. Davidian and R. J. Carroll. *Variance Function Estimation*. Journal of the American Statistical Association, 1987. Cited on page 4.

[3] Julian Faraway. *Linear Models With R*. CRC, 2009. Cited on page 4.

[4] Andrzej Galecki and Tomasz Burzykowski. *Linear Mixed-Effects Models Using R: A Step-by-Step Approach*. Springer, 2013. Cited on pages 4 and 10.

[5] R. C. Geary. *The Ratio of the Mean Deviation to the Standard Deviation as a Test of Normality*. Biometrika, 1935. Cited on page 7.

[6] E. B. Wilson and M. M. Hilferty. *The distribution of chi-squared*. Proceedings of the National Academy of Science, 1931. Cited on page 8.