

Regression III

More Flexible Fitting Methods

Dave Armstrong

Goals for Today

1. Discuss methods for more flexible fitting.
 - Classification and Regression Trees (CART)
 - Random Forests Regression
 - Multivariate Adaptive Regression Splines (MARS)
 - Adaptive LASSO with Polynomial Expansion (Polywog)
2. Discuss MARS in an inferential context.

2 / 117

Classification and Regression Trees (CART)

CART works in a decision-tree framework.

- Considering all independent variables, which dichotomization on one of them explains the most variance.
- Conditional on the previous *split*, which next dichotomization explains the most variance.
- Loss function is the well-known residual sum of squares.
- Continue until some stopping rule is met.

Notes

Type notes here...

3 / 117

4 / 117

Notation

$$f(X_i) = T(X_i, \Theta) \equiv \sum_{b=1}^B c_b I(X_i \in R_b)$$

- $T()$ is a regression tree, with rules Θ regarding tree depth, stopping rules, etc...
- X_i is the data.
- c_b is the predicted value in each of the B regions.
- $I()$ is an indicator function.
- R_b defines the different regions in the space.

5 / 117

Notes

Type notes here...

6 / 117

Stopping Rules

- Candidate splits must increase R^2 by a pre-specified amount (the `cp` parameter, default=.01).
- Each candidate split must have at least `minsplit` observations in it (default=20).
- Each terminal node must have at least `minbucket` observations in it. Defaults to `round(minsplit/3)`.
- Tree depth - starting with the root node (0), what is the maximum depth of any node (defaults to 30).

7 / 117

Notes

Type notes here...

8 / 117

Example

```
library(rpart)
library(dplyr)
library(car)
data(SLID)
SLID <- SLID %>%
  dplyr::select(wages, age, education) %>%
  na.omit
mod <- rpart(log(wages) ~ age + education, data=SLID)
mod
```

Notes

Type notes here...

9 / 117

10 / 117

Decision Tree

```
plot(mod)
text(mod)
```

Notes

Type notes here...

11 / 117

12 / 117

Surface Plot

```
age.s <- seq(20, 95, length=25)
educ.s <- seq(9, 26, length=25)
cartpred <- function(x,y){
  predict(mod,
    newdata=data.frame(
      age = x,
      education=y))
}
p <- outer(age.s,
  educ.s,
  cartpred)

library(plotly)
plot_ly() %>%
  add_surface(x=age.s, y=-educ.s, z=-exp(p)) %>
  layout(
    scene= list(
      xaxis=list(title="Age"),
      yaxis=list(title="Education"),
      zaxis=list(title="Predicted Wage")))

```



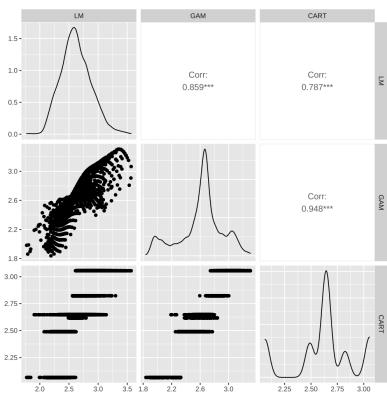
Notes

Type notes here...

13 / 117

14 / 117

Compare to Other Models



15 / 117

Notes

Type notes here...

16 / 117

Smoothness of CART Models

Problems with CART

- Not particularly smooth - step functions aren't great approximations for smooth curves (though they can do OK).
- No real means for inference here. Bootstrapping can be problematic because the function is "non-regular" (small data changes can result in wild changes in the model)
- In more complicated models, it's difficult to figure out what effects look like.

17 / 117

Notes

Type notes here...

18 / 117

Visualizing Partial Effects: Partial Dependence Plot

The PDP plots the change in the average predicted value for a subset of features S , averaged over the subset of features C , where C is the complement of S . Formally:

$$f_S = \mathbb{E}_{x_C} [f(\mathbf{x}_S, \mathbf{x}_C)] = \int f(\mathbf{x}_S, \mathbf{x}_C) dP(\mathbf{x}_C)$$

In words: we are predicting $f()$ with the variables in S averaged over all of the variables in C .

19 / 117

Notes

Type notes here...

20 / 117

ICE Plots

ICE disaggregates the PDP.

- The PDP is obtained by averaging over all of the ICE curves.
- Plots N different curves to enable evaluation of effect heterogeneity.
- Heterogeneity essentially means interactions with variables in C .

$$f_{S_i} = \mathbb{E}_{x_{C_i}} [f(\mathbf{x}_S, \mathbf{x}_{C_i})]$$

21 / 117

Notes

Type notes here...

22 / 117

Ice Ice Baby

```
library(ICEbox)
library(RColorBrewer)
ice1 <- ice(mod, SLID, y=mod$y,
            predictor="age")
cice1 <- clusterICE(ice1, nClusters=3,
                      plot.legend=TRUE,
                      colorvec=brewer.pal(3,
"Set1"))
```

Prop.

23 / 117

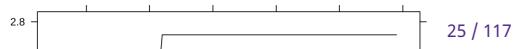
Notes

Type notes here...

24 / 117

PDPs in R

```
library(pdp)
p1 <- partial(mod, pred.var="age", chull=TRUE)
plotPartial(p1)
```



Notes

Type notes here...

26 / 117

Ensemble Methods

Ensemble methods produce a bunch of (M) trees to better fit $f(X)$ and to prevent overfitting, with general form:

$$f(X_i) = \sum_{m=1}^M T_m(X_i, \Theta_m)$$

Tree Bagging (Bootstrap Aggregating)

- Draw lots of random samples from the data
- Fit a *deep* tree to each random sample.
- Average across the trees $\hat{f}_{\text{bag}}(X_i) = \frac{1}{M} \sum_{m=1}^M T_m(X_i, \Theta_m)$

Depends on the often dubious assumption of independence across trees to reduce bias and variance.

27 / 117

Notes

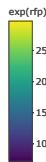
Type notes here...

28 / 117

Random Forests

A tree-bagging algorithm meant to increase independence across trees.

- In each random sample, only a small random subset (a) of the total j covariates is used in the splitting algorithm.
- Reduces bias and variance in the aggregate when a is small.



```
library(randomForest)
rfmod <- randomForest(log(wages) ~ age+education,
                      data=SLID,
                      type="regression")
```

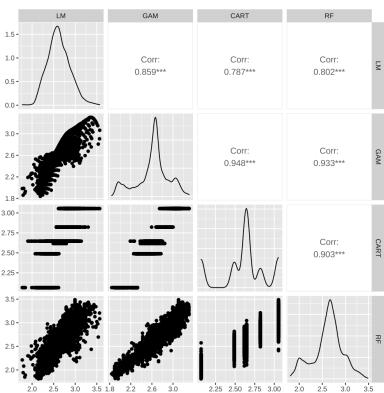
29 / 117

Notes

Type notes here...

30 / 117

Compare to Other Models



31 / 117

Notes

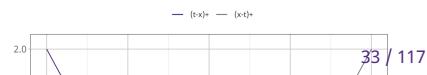
Type notes here...

32 / 117

Multivariate Adaptive Regression Splines (MARS)

The main component of MARS is a pair of piecewise linear (hinge) splines.

$$(x - t)_+ = \begin{cases} x - t & \text{if } x > t \\ 0 & \text{otherwise.} \end{cases}$$
$$(t - x)_+ = \begin{cases} t - x & \text{if } x < t \\ 0 & \text{otherwise.} \end{cases}$$



Notes

Type notes here...

34 / 117

MARS Notation

MARS takes the form:

$$f(x) = \beta_0 + \sum_{m=1}^M \beta_m h_m(x)$$

where h_m is the pair of hinge functions.

Computationally:

- Forward pass - add pairs of hinge functions by reduction in SSRes until all pairs are in.
- Backward pass - take individual functions out by min increase in SSRes until GCV criterion is satisfied.

35 / 117

Notes

Type notes here...

36 / 117

Interactions

- The `degree` parameter in the R algorithm controls the degree of interaction you want to allow.
- This can make the model really complicated because it's expanding all possible interactions among hinge functions and then pulling them out on the backward pass step.
- This model is more easily constrained (particular w.r.t additivity) than the other models we talked about before.
- You can also identify variables that will enter the model linearly *if they enter the model at all*.

37 / 117

Notes

Type notes here...

38 / 117

MARS Wages

```
library(earth)
emod <- earth(log(wages) ~ age +
  education, data=SLID,
  degree=3)

summary(emod)

## Call: earth(formula=log(wages)-age+education, data=SLID, degree=3)
##
##          coefficients
##  (Intercept) 2.104825097
##  h(32-age) -0.049452115
##  h(age-32) -0.03079281
##  h(12.6-education) -0.16237528
##  h(education-12.6) 0.15308353
##  h(age-32) * h(education-17.1) -0.01150617
##  h(age-32) * h(17.1-education) 0.00666845
##  h(age-34) * h(education-12.6) 0.00552501
##  h(51-age) * h(education-12.6) -0.00489692
##  h(58-age) * h(12.6-education) 0.00426955
##  h(age-58) * h(12.6-education) -0.01536656
##
##  Selected 11 of 12 terms, and 2 of 2 predictors
##  Termination condition: RSS changed by less than 0.001 at 12 terms
##  Importance: age, education
##  Number of terms at each degree of interaction: 1 4 6
##  GCV 0.166029   RSS 657.835   GRSq 0.3457333   RSq 0.3538597
```

39 / 117

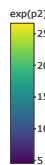
Notes

Type notes here...

40 / 117

Surface Plot

```
marspred <- function(x,y){  
  predict(emod,  
         newdata=data.frame(age = x,  
                           education=y))  
}  
p2 <- outer(age.s,  
            educ.s,  
            marspred)  
  
plot_ly() %>%  
  add_surface(x=-age.s, y=-educ.s, z=-exp(p2)) %>%  
  layout(  
    scene=list(  
      xaxis=list(title="Age"),  
      yaxis=list(title="Education"),  
      zaxis=list(title="Predicted Wage")  
    ))
```



Notes

Type notes here...

41 / 117

42 / 117

Plots

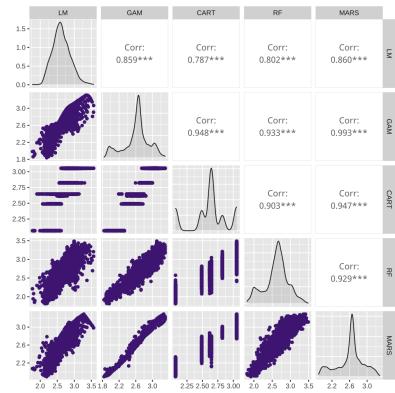
Notes

Type notes here...

43 / 117

44 / 117

Compare to Other Models



45 / 117

Notes

Type notes here...

Variance Models

- You can't get confidence intervals from these models because they don't take into account the selection mechanism.
- MARS picks values essentially because they are good predictors, so the items in the model will necessarily have small p-values.
- You can get prediction intervals for the - essentially the variability in future observations predicted by the model.
- The `varmod.method` allows you to model the residual variance by modeling the absolute value of the residuals as a function of the fitted values.
- Prediction variance is:

$$\varepsilon_{i,future}^2 = \frac{(y_i - \hat{y}_i)^2}{(1 - h_{ii})} + \text{modvar}_i$$

47 / 117

46 / 117

Notes

Type notes here...

48 / 117

Prediction Variances in earth

```
library(mgcv)
e2 <- earth(log(wages) ~ age + education,
  data=SLID,
  nfold=10, ncross=10, pmethod="cv",
  degree=2, varmod.meth="gam")
plotmo(e2, pt.col=1, level=.95)
```

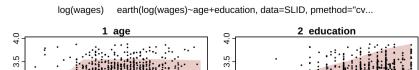
49 / 117

Notes

Type notes here...

50 / 117

Plots



51 / 117

Notes

Type notes here...

52 / 117

Polywog

Polywog is a method developed by Kenkel and Signorino which puts two pieces we've already considered together:

- Polynomial expansion: If the degree = 3 and we have variables $\{x_1, x_2\}$ in our model, then the following terms would be included in the expansion:
 $x_1, x_2, x_1^2, x_2^2, x_1^3, x_2^3, x_1x_2, x_1^2x_2, x_2^2x_1$.
- Adaptive Lasso: We use the adaptive LASSO to figure out which of the polynomial expansion terms to keep in the model.

53 / 117

Notes

Type notes here...

54 / 117

Polywog Example

```
library(polywog)
p1 <- polywog(log(wages) ~ age +
  education, data=SLID, degree=4)

## Call:
## polywog(formula = log(wages) ~ age + education, data = SLID,
##          degree = 4)

## Coefficients:
##                               Estimate Std. Error
## (Intercept)           1.438e+00   NA
## age                  1.830e-02   NA
## education            0.000e+00   NA
## age^2                 0.000e+00   NA
## age.education        4.063e-03   NA
## education^2          -4.891e-03  NA
## age^3                 0.000e+00   NA
## age^2.education      -7.456e-05  NA
## age.education^2       1.129e-04  NA
## education^3           5.197e-05  NA
## age^4                 1.172e-08  NA
## age^3.education      0.000e+00   NA
## age^2.education^2    0.000e+00   NA
## age.education^3      0.000e+00   NA
## education^4           0.000e+00   NA
##
## Regularization method: Adaptive LASSO
## Adaptive weights: inverse linear model coefficients
## Number of observations: 4014
## Polynomial expansion degree: 4
## Model family: gaussian
## Bootstrap iterations: 0
```

55 / 117

Notes

Type notes here...

56 / 117

Plots

57 / 117

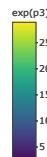
Notes

Type notes here...

58 / 117

Surface Plot

```
pwogpred <- function(x,y){  
  predict(p1,  
    newdata=data.frame(age = x,  
                      education=y))  
}  
p3 <- outer(age.s,  
            educ.s,  
            pwogpred)  
  
plot_ly() %>%  
  add_surface(x=-age.s, y=-educ.s, z=-exp(p3)) %>%  
  layout(  
    scene= list(  
      xaxis=list(title="Age"),  
      yaxis=list(title="Education"),  
      zaxis=list(title="Predicted Wage")  
    ))
```



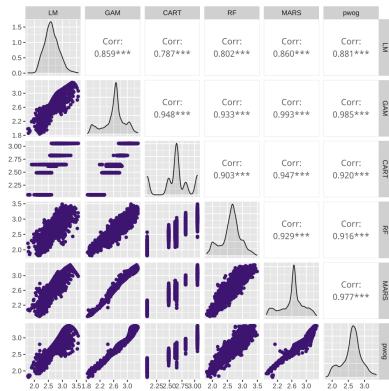
59 / 117

Notes

Type notes here...

60 / 117

Compare to Other Models



61 / 117

Notes

Type notes here...

62 / 117

Barry and Kleinberg Data

```
load(file("https://quantoid.net/files/reg3/bk.rda"))
bk <- as.data.frame(bk)
model2 <- gamlss(usfdi2000_adj ~ L_ab_sum + L_hse_usnum + L_hse_sanction +
  L_growth + L_lnpercap + L_lnpop + L_polity2 + L_durable +
  L_civintensity + L_spending + L_s_us + L_lnustrade +
  L_us_distance + L_usstock2000_adj + allfdi2000, data = bk)
```

```
## GAMLSS-RS iteration 1: Global Deviance = 16189.29
## GAMLSS-RS iteration 2: Global Deviance = 16189.29
```

```
model3 <- gamlss(usfdi2000_adj ~ L_sanmeanshare + L_tradeshare + L_hse_sanction +
  L_growth + L_lnpercap + L_lnpop + L_polity2 + L_durable +
  L_civintensity + L_spending + L_s_us + L_lnustrade +
  L_us_distance + L_usstock2000_adj + allfdi2000, data = bk)
```

```
## GAMLSS-RS iteration 1: Global Deviance = 16182.26
## GAMLSS-RS iteration 2: Global Deviance = 16182.26
```

63 / 117

Notes

Type notes here...

64 / 117

Cart Models

```
bk.cart1 <- rpart(usfdi2000_adj ~ L_ab_sum + L_hse_usnum + L_hse_sanction +
  L_growth + L_lnpercap + L_lnpop + L_polity2 + L_durable +
  L_civintensity + L_spending + L_s_us + L_lnustrade +
  L_us_distance + L_usstock2000_adj + allfdi2000, data = bk)
bk.cart2 <- rpart(usfdi2000_adj ~ L_sancmeanshare + L_tradeshare + L_hse_sanction +
  L_growth + L_lnpercap + L_lnpop + L_polity2 + L_durable +
  L_civintensity + L_spending + L_s_us + L_lnustrade +
  L_us_distance + L_usstock2000_adj + allfdi2000, data = bk)
```

65 / 117

Notes

Type notes here...

66 / 117

CART Model Results

```
bk.cart1
      ## n= 2863
      ## 
      ## node), split, n, deviance, yval
      ##   * denotes terminal node
      ##
      ## 1) root 2863 61629.460 2.976535
      ## 2) L_usstock2000_adj< 8.908139 2184 36891.230 1.973858
      ##   4) L_usstock2000_adj< 6.56151 1169 12158.840 1.038018 *
      ##   5) L_usstock2000_adj>=6.56151 1015 22529.440 3.051688
      ##   10) L_growth< 2.675 330 8621.694 1.847877 *
      ##   11) L_growth>=2.675 685 13199.140 3.631627 *
      ## 3) L_usstock2000_adj>=8.908139 679 15480.040 6.201639
      ##   6) L_usstock2000_adj< 10.03128 327 8397.456 4.996460 *
      ##   7) L_usstock2000_adj>=10.03128 352 6166.405 7.321224 *
```

67 / 117

Notes

Type notes here...

68 / 117

Random Forests

```
bk.cart1 <- rpart(usfdi2000_adj ~ L_ab_sum + L_hse_usnum + L_hse_sanction +
  L_growth + L_lnpercap + L_lnpop + L_polity2 + L_durable +
  L_civintensity + L_spending + L_s_us + L_lnustrade +
  L_us_distance + L_usstock2000_adj + allfdi2000, data = bk)
bk.cart2 <- rpart(usfdi2000_adj ~ L_sancmeanshare + L_tradeshare + L_hse_sanction +
  L_growth + L_lnpercap + L_lnpop + L_polity2 + L_durable +
  L_civintensity + L_spending + L_s_us + L_lnustrade +
  L_us_distance + L_usstock2000_adj + allfdi2000, data = bk)
```

69 / 117

Notes

Type notes here...

70 / 117

Random Forests

```
bk.rf1 <- randomForest(usfdi2000_adj ~ L_ab_sum + L_hse_usnum + L_hse_sanction +
  L_growth + L_lnpercap + L_lnpop + L_polity2 + L_durable +
  L_civintensity + L_spending + L_s_us + L_lnustrade +
  L_us_distance + L_usstock2000_adj + allfdi2000,
  data = bk, mtry=3)
bk.rf2 <- randomForest(usfdi2000_adj ~ L_sancmeanshare + L_tradeshare + L_hse_sanction +
  L_growth + L_lnpercap + L_lnpop + L_polity2 + L_durable +
  L_civintensity + L_spending + L_s_us + L_lnustrade +
  L_us_distance + L_usstock2000_adj + allfdi2000,
  data = bk, mtry=3)
```

71 / 117

Notes

Type notes here...

72 / 117

Variable Importance

```
imp1 <- importance(bk.rf1)
imp1/max(imp1)
```

```
imp2 <- importance(bk.rf2)
imp2/max(imp2)
```

```
##          IncNodePurity
## L_ab_sum      0.13914331
## L_hse_usnum    0.27336546
## L_hse_sanction  0.02918143
## L_growth       0.13460446
## L_lnperscap     0.57810416
## L_lnpop        0.47848665
## L_polity2       0.23685643
## L_durable        0.39365502
## L_civintensity   0.06230792
## L_spending        0.37971906
## L_s_us           0.37797556
## L_lnustrade      0.70098276
## L_us_distance     0.24532186
## L_usstock2000_adj 1.00000000
## L_usstock2000e_adj 0.33512000
## allfdi2000      0.33512007
```

73 / 117

Notes

Type notes here...

74 / 117

ICE Plot

```
library(RColorBrewer)
bk.i <- ice(bk.rf1, bk,
            predictor = "L_hse_usnum",
            frac_to_build = 1)
crv <- bk.i$curve
crv <- t(apply(crv, 1, function(x)x-mean(x)))
sapply(1:10, function(i){
  k <- kmeans(crv, centers=i)
  k$betweenss/(k$tot.withinss+k$betweenss)
})
bk.c <- clusterICE(bk.i, nClusters = 6,
                     colorvec=brewer.pal(6, "Set1"),
                     plot.legend = TRUE)
```

Prop.

75 / 117

Notes

Type notes here...

76 / 117

ICE Plot

```
bk.i2 <- ice(bk.rf2, bk,
  predictor = "L_sancmeanshare",
  frac_to_build = 1,
  num_grid_pts = 25)
crv <- bk.i2$ice_curves
crv <- t(apply(crv, 1, function(x)x-mean(x)))
sapply(2:10, function(i){
  k <- kmeans(crv, centers=i)
  ksbetweens/(k$tot.withins+k$betweens)
})
bk.c <- clusterICE(bk.i2, nClusters = 5,
  colorvec=brewer.pal(5, "Set1"),
  plot_legend = TRUE)
```

2 Prop.

77 / 117

Notes

Type notes here...

78 / 117

Random Forests

```
bk.e1 <- earth(usfdi2000_adj ~ L_ab_sum + L_hse_usnum + L_hse_sanction +
  L_growth + L_lnpercap + L_lnpop + L_polity2 + L_durable +
  L_civintensity + L_spending + L_s_us + L_lnustrade +
  L_us_distance + L_usstock2000_adj + allfdi2000,
  data = bk, degree=3, pmethod="backward")
bk.e2 <- earth(usfdi2000_adj ~ L_sancmeanshare + L_tradeshare + L_hse_sanction +
  L_growth + L_lnpercap + L_lnpop + L_polity2 + L_durable +
  L_civintensity + L_spending + L_s_us + L_lnustrade +
  L_us_distance + L_usstock2000_adj + allfdi2000,
  data = bk, degree=3, pmethod="backward")
```

79 / 117

Notes

Type notes here...

80 / 117

Results 1

```
summary(bk.e1)

## Call: earth(formula=usfdi2000_adj~L_ab_sum*L_hse_usnum+L_hse_sanction+...),
##           data=bk, pmethod="backward", degree=3)
##
##          coefficients
## (Intercept)          0.63392319
## h(17-L_hse_usnum)      0.24534151
## h(L_hse_usnum-17)     0.14932381
## h(5.84698-L_usstock2000_adj) -0.11501938
## h(L_usstock2000_adj-5.84698)  1.3988457
## h(17-L_hse_usnum) + h(L_s_us-0.452087) -2.07349368
## h(17-L_hse_usnum) + h(L_lnpercap-8.72583) + h(L_s_us-0.452087)  1.32325615
## h(17-L_hse_usnum) + h(8.72583-L_lnpercap) + h(L_s_us-0.452087)  0.60472631
## h(17-L_hse_usnum) + h(L_lnpop-16.1764) + h(L_s_us-0.452087)  0.71042312
## h(17-L_hse_usnum) + h(16.1764-L_lnpop) + h(L_s_us-0.452087)  0.63388703
## h(17-L_hse_usnum) + h(0.452087-L_s_us) + h(6.39192-L_lntrade) -0.00000000
## h(17-L_hse_usnum-17) + h(0.593291-L_s_us) + h(L_lntrade-9.24532) -0.51248289
## h(L_hse_usnum-17) + h(0.593291-L_s_us) + h(L_lntrade-9.24532) -0.00992691
## h(13-L_growth) + h(0.17926-L_s_us) + h(L_usstock2000_adj-5.84698)  0.05057111
## h(13-L_growth) + h(0.49552-L_lnpercap) + h(L_usstock2000_adj-5.84698) -0.04877511
## h(13-L_growth) + h(L_inpop-18.6438) + h(L_usstock2000_adj-5.84698)  0.18606737
## h(13-L_growth) + h(18.6438-L_inpop) + h(L_usstock2000_adj-5.84698) -0.00737439
## h(L_growth-7.84) + h(-9-L_polity2) + h(L_usstock2000_adj-5.84698) -0.18431613
## h(13-L_growth) + h(21.5-L_spending) + h(L_usstock2000_adj-5.84698)  0.00391730
## h(13-L_growth) + h(L_usstock2000_adj-5.84698) + h(59834.9-allfdi2000) -0.00000080
## 
## Selected 20 of 31 terms, and 10 of 15 predictors
## Termination condition: Reached nk 31
```

81 / 117

Notes

Type notes here...

82 / 117

Results 2

```
summary(bk.e2)

## Call: earth(formula=usfdi2000_adj~L_sancmeanshare+L_tradeshare+L_hse_s...,,
##           data=bk, pmethod="backward", degree=3)
##
##          coefficients
## (Intercept)          1.2599580
## h(L_s_us-0.759834)  23.4287388
## h(5.84698-L_usstock2000_adj) -0.1385368
## h(L_usstock2000_adj-5.84698)  1.4179459
## h(13-L_growth) + h(L_usstock2000_adj-5.84698)  0.06533956
## h(L_lnpercap-10.8476) + h(6.759834-L_s_us)  8.1232693
## h(9.17926-L_intrade) + h(L_usstock2000_adj-5.84698)  0.5108149
## h(L_lntrade-6.17028) + h(L_usstock2000_adj-5.84698) -0.2692848
## h(13-L_growth) + h(9.4572-L_lnpercap) + h(L_usstock2000_adj-5.84698) -0.0572491
## h(13-L_growth) + h(L_inpop-18.6438) + h(L_usstock2000_adj-5.84698)  0.1023436
## h(13-L_growth) + h(18.6438-L_inpop) + h(L_usstock2000_adj-5.84698) -0.0263415
## h(13-L_growth) + h(16.9-L_spending) + h(L_usstock2000_adj-5.84698)  0.0032823
## h(1.39-L_growth) + h(L_lntrade-8.17926) + h(L_usstock2000_adj-5.84698)  0.0617883
## h(L_growth-1.39) + h(L_lntrade-8.17926) + h(L_usstock2000_adj-5.84698)  0.0421413
## h(13-L_growth) + h(L_usstock2000_adj-5.84698) + h(59834.9-allfdi2000) -0.0000012
## h(L_lnpercap-10.3123) + h(L_lntrade-8.17926) + h(L_usstock2000_adj-5.84698) -1.8565817
## h(-9-L_polity2) * h(8.17926-L_intrade) + h(L_usstock2000_adj-5.84698) -0.9600373
## h(0.33333-L_s_us) * h(8.17926-L_intrade) + h(L_usstock2000_adj-5.84698) -2.9565317
## h(L_s_us-0.33333) * h(8.17926-L_intrade) + h(L_usstock2000_adj-5.84698) -1.6569208
## h(8.17926-L_intrade) * h(5425-L_us_distance) + h(L_usstock2000_adj-5.84698)  0.000112
## 
## Selected 20 of 31 terms, and 10 of 15 predictors
## Termination condition: Reached nk 31
```

83 / 117

Notes

Type notes here...

84 / 117

ICE Plot

```
bk.i <- ice(bk.e1, bk,
            predictor = "L_hse_usnum",
            frac_to_build = 1)
crv <- bk$ice_curves
crv <- t(apply(crv, 1, function(x)x-mean(x)))
apply(2:10, function(i){
  k <- kmeans(crv, centers=i)
  k$betweenss/(k$tot.withins+k$betweenss)
})
bk.c <- clusterICE(bk.i, nClusters = 4,
                     colorvec=brewer.pal(4, "Set1"),
                     plot_legend = TRUE)
```

Prop.

85 / 117

Notes

Type notes here...

86 / 117

Investigating clusters

```
library(nnet)
bk$cluster <- bk$c$cluster
clust.mod <- multinom(cluster ~ L_ab_sum +
  L_hse_usnum + L_hse_sanction +
  L_growth + L_lnpercap + L_lnpop +
  L_polity2 + L_durable + L_civintensity +
  L_spending + L_s_us + L_lnustrade +
  L_us_distance + L_usstock2000_adj +
  allfdi2000, data=bk)

## # weights: 68 (48 variable)
## initial value 3968.960756
## iter  10 value 3516.858960
## iter  20 value 3460.161139
## iter  30 value 3448.755375
## iter  40 value 3361.707671
## iter  50 value 3283.336646
## final value 3281.728662
## converged
```

```
DAMisc::mnlChange(clust.mod, bk)
```

	[,1]	[,2]	[,3]	[,4]
## L_ab_sum	-0.007	-0.105	-0.058*	0.170
## L_hse_usnum	-0.049*	-0.047*	0.043*	0.054*
## L_hse_sanction	0.013*	0.009*	-0.039*	-0.001
## L_growth	0.113*	-0.149*	0.011*	-0.041*
## L_lnpercap	-0.268*	0.141*	-0.245*	-0.099*
## L_lnpop	-0.299*	0.030*	0.276*	-0.017*
## L_polity2	0.099*	-0.016*	-0.006*	0.012*
## L_durable	0.250*	-0.115*	0.008*	-0.141*
## L_civintensity	0.114*	-0.086*	-0.022*	-0.007*
## L_spending	-0.221*	-0.018	0.181*	0.058*
## L_s_us	-0.152*	-0.082*	0.071*	0.084*
## L_lnustrade	0.294*	0.205*	-0.146*	-0.354*
## L_us_distance	0.228*	-0.177*	-0.006	-0.045
## L_usstock2000_adj	0.191*	-0.091*	-0.395*	0.296*
## allfdi2000	0.060	0.042	-0.016	-0.086*

87 / 117

Notes

Type notes here...

88 / 117

Venus

Venus is a project that I am working on with Duncan Murdoch.

- Some variables are subject to a MARS fit.
 - Good way to control for variables.
- Other variables are included in their assumed parametric form.

89 / 117

Notes

Type notes here...

90 / 117

Details

$$\mathbf{y} = \alpha + \mathbf{X}\beta + \mathbf{Z}\Gamma + \varepsilon$$

We create $e^{(y)}$ with MARS:

$$\mathbf{y} = \lambda H(\mathbf{Z}) + e^{(y)}$$

and $\mathbf{e}^{(x)}$ with

$$\mathbf{X} = \theta H(\mathbf{Z}) + \mathbf{e}^{(X)}$$

Then we regress $e^{(y)}$ on $\mathbf{e}^{(X)}$ to obtain estimates of β controlling for \mathbf{Z} in a flexible way.

91 / 117

Notes

Type notes here...

92 / 117

In R

```
remotes::install_github("dmurdoch/venus")  
  
library(venus)  
bk.v1 <- venus(usfdi2000_adj ~ L_ab_snum + L_hse_usnum,  
                 usfdi2000_adj ~ L_hse_sanction +  
                 L_growth + L_lnpercap + L_lnpop + L_polity2 + L_durable +  
                 L_civintensity + L_spending + L_s_us + L_lnustrade +  
                 L_us_distance + L_usstock2000_adj + allfdi2000,  
                 data = bk)  
bk.v2 <- venus(usfdi2000_adj ~ L_sancmeanshare + L_tradeshare,  
                 usfdi2000_adj ~ L_hse_sanction +  
                 L_growth + L_lnpercap + L_lnpop + L_polity2 + L_durable +  
                 L_civintensity + L_spending + L_s_us + L_lnustrade +  
                 L_us_distance + L_usstock2000_adj + allfdi2000,  
                 data = bk)
```

93 / 117

Notes

Type notes here...

94 / 117

Summary

```
summary(bk.v1$mainFit)                                summary(bk.v2$mainFit)  
  
## Call:  
## lm(formula = yResids ~ mainModelmatrix)  
##  
## Residuals:  
##   Min     1Q Median     3Q    Max  
## -16.664 -1.225  1.323  2.558  9.208  
##  
## Coefficients:  
##             Estimate Std. Error t value Pr(>|t|)          Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -1.170e-16 7.493e-02  0.000 1.000000  ## (Intercept) -9.826e-17 7.508e-02  0.000  1.000  
## mainModelmatrix_ab_snum 2.330e-02 3.994e-02  0.583 0.559609  ## mainModelmatrix_sancmeanshare 5.799e-02 5.111e-02 1.135  0.257  
## mainModelmatrix_hse_usnum -5.278e-02 1.433e-02 -3.684 0.000234 ***  ## mainModelmatrix_tradeshare -9.313e-03 8.569e-03 -1.087  0.277  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
## Residual standard error: 4.009 on 2860 degrees of freedom  
## Multiple R-squared:  0.004733, Adjusted R-squared:  0.0004037  
## F-statistic: 6.881 on 2 and 2860 DF, p-value: 0.001131
```

95 / 117

Notes

Type notes here...

96 / 117

In GAMLSS

```
remotes::install_url("https://quantoid.net/files/gamlss.add2_1.0-0.tar.gz")

library(gamlss)
library(gamlss)

bk.g1 <- gamlss(usfdi2000_adj ~ L_ab_sum + L_hse_usnum +
  tr(~L_hse_sanction + L_growth + L_lpercap +
  L_lnpop + L_polity2 + L_durable + L_civintensity +
  L_spending + L_s_us + L_lustrade +
  L_us_distance + L_usstock2000_adj + alldfi2000),
  data = bk, trace=FALSE, control=gamlss.control(n.cyc=100))

## GAMLSS-RS iteration 1: Global Deviance = 16188.54
## GAMLSS-RS iteration 2: Global Deviance = 16188.54

bk.g2 <- gamlss(usfdi2000_adj ~ L_sancmeanshare + L_tradshare +
  tr(~L_hse_sanction + L_growth + L_lpercap +
  L_lnpop + L_polity2 + L_durable + L_civintensity +
  L_spending + L_s_us + L_lustrade +
  L_us_distance + L_usstock2000_adj + alldfi2000),
  data = bk, trace=FALSE)
```

97 / 117

Notes

Type notes here...

Summary

```
summary(bk.g1)                                     summary(bk.g2)

## ****
## Family: c("NO", "Normal")
## Call: gamlss(formula = usfdi2000_adj ~ L_ab_sum + L_hse_usnum +
##   tr(~L_hse_sanction + L_growth + L_lpercap +
##   L_lnpop + L_polity2 + L_durable + L_civintensity +
##   L_spending + L_s_us + L_lustrade +
##   L_us_distance + L_usstock2000_adj + alldfi2000),
##   data = bk, control = gamlss.control(n.cyc = 100),
##   trace = FALSE)
## Fitting method: RS()
## -----
## Mu link function: identity
## Mu Coefficients:
## (Intercept) Estimate Std. Error t value Pr(>|t|)
## 3.25020 0.19402 16.75 <2e-16 ***
## L_ab_sum -0.05211 0.04071 1.28 0.2006
## L_hse_usnum -0.01590 0.00935 -1.70 0.0992 .
## -
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## -----
## Sigma link function: log
## Sigma Coefficients:
## Estimate Std. Error t value Pr(>|t|)
```

99 / 117

Notes

Type notes here...

98 / 117

100 / 117

MARS

```
library(gamlss.add2)
bk.g3 <- gamlss(usfdi2000_adj ~ L_ab_sum + L_hse_usnum +
  ma(~L_hse_sanction + L_growth + L_lnpercap +
  L_lnpop + L_polity2 + L_durable + L_civintensity +
  L_spending + L_s_us + L_lnustrade +
  L_us_distance + L_usstock2000_adj) + allfdi2000),
data = bk, trace=FALSE, control=gamlss.control(n.cyc=100))
```

```
## GAMLSS-RS iteration 1: Global Deviance = 16051.26
## GAMLSS-RS iteration 2: Global Deviance = 16057.73
## GAMLSS-RS iteration 3: Global Deviance = 16057.73
```

```
bk.g4 <- gamlss(usfdi2000_adj ~ L_sancmeanshare + L_tradeshare +
  ma(~L_hse_sanctid + L_growth + L_lnpercap +
  L_lnpop + L_polity2 + L_durable + L_civintensity +
  L_spending + L_s_us + L_lnustrade +
  L_us_distance + L_usstock2000_adj) + allfdi2000),
data = bk, trace=FALSE)
```

101 / 117

Notes

Type notes here...

Summary

```
summary(bk.g3)
```

```
summary(bk.g4)

## ****
## Family: c("NO", "Normal")
## Call: gamlss(formula = usfdi2000_adj ~ L_sancmeanshare +
##   ma(~L_hse_sanction + L_growth + L_lnpercap +
##   L_lnpop + L_polity2 + L_durable + L_civintensity +
##   L_spending + L_s_us + L_lnustrade +
##   L_us_distance + L_usstock2000_adj) + allfdi2000),
##   data = bk, control = gamlss.control(n.cyc = 100),##
##   trace = FALSE)
## Fitting method: RS()
## -----
## Mu link function: identity
## Mu Coefficients:
## (Intercept) Estimate Std. Error t value Pr(>|t|)
## 4.564214 0.189639 24.068 <2e-16 ***
## L_ab_sum 0.010584 0.039788 0.266 0.79
## L_hse_usnum -0.082545 0.005139 -9.032 <2e-16 ***
## -
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## -----
## Sigma link function: log
## Sigma Coefficients:
## Estimate Std. Error t value Pr(>|t|)
```

103 / 117

Notes

Type notes here...

102 / 117

104 / 117

Tests

```
VC.test(model2, bk.g1)

## Vuong's test: -0.733 it is not possible to discriminate between models: model2 and bk.g1
## Clarke's test: 1381 p-value= 0.8616 it is not possible to discriminate between models: model2 and bk.g1

VC.test(bk.g1, bk.g3)

## Vuong's test: -2.898 model bk.g3 is preferred over bk.g1
## Clarke's test: 1232 p-value= 0 bk.g3 is preferred over bk.g1

VC.test(model3, bk.g2)

## Vuong's test: 0.44 it is not possible to discriminate between models: model3 and bk.g2
## Clarke's test: 1474 p-value= 0.1164 it is not possible to discriminate between models: model3 and bk.g2

VC.test(bk.g2, bk.g4)

## Vuong's test: -3.37 model bk.g4 is preferred over bk.g2
## Clarke's test: 1214 p-value= 0 bk.g4 is preferred over bk.g2
```

105 / 117

Notes

Type notes here...

106 / 117

Inference

Venus has good inferential properties.

- Bias, MSE and CI coverage errors go to 0 as N increases.
- Dominates a naive linear model in all but the perfect additive linear case.

Other models:

- The venus result suggests that the GAMLSS model should have decent inferential properties on the parametric terms.
- Trees, MARS and Polywog would need data splitting or something similar to make appropriate inferences.
 - Model building could be done on a training sample and then the appropriate model could be estimated on the other half of the data.

107 / 117

Notes

Type notes here...

108 / 117

Mars Example with Inference

```
set.seed(734)
train.samp <- sample(1:nrow(bk),
                     floor(nrow(bk)*.6),
                     replace=FALSE)
test.samp <- setdiff(1:nrow(bk), train.samp)
bk.e1 <- earth(usfdi2000_adj ~ L_lab.sum + L_hse.usnum + L_hse_sanction +
  L_growth + L_lpercap + L_lnpo + L_polity2 + L_durable +
  L_civintensity + L_spending + L_s_us + L_lnustrade +
  L_us_distance + L_usstock2000_adj + allfdi2000,
  data = bk[train.samp, ], degree=3, pmethod = "backward")
```

Generating predictions:

```
h <- function(x){
  out <- eval(x)
  out <- out*(out > 0)
  out
}
X <- model.matrix(bk.e1)
hinges <- colnames(X)[-1]
hinges <- gsub("x", "n", hinges, fixed=TRUE)
form <- reformat(hinges, response="usfdi2000_adj")
lmod1 <- lm(form, data=bk[test.samp, ])
lmod2 <- lm(form, data=bk[train.samp, ])
```

109 / 117

Notes

Type notes here...

110 / 117

Summary

```
summary(lmod1)

##
## Call:
## lm(formula = form, data = bk[test.samp, ])
##
## Residuals:
##   Min     1Q   Median     3Q    Max 
## -15.754 -1.439  1.442  2.656 10.274 
## 
## Coefficients:
##             Estimate
## (Intercept) 3.269e-01
## h(L_usstock2000_adj - 6.53771) 6.859e-01
## h(6.53771 - L_usstock2000_adj) -1.476e-01
## h(L_hse.usnum - 17) 3.786e-01
## h(17 - L_hse.usnum) 1.763e-01
## h(L_lnpo - 16.6189) 3.585e-01
## h(L_hse.usnum - 20) -2.383e-01
## h(L_usstock2000_adj - 6.53771):h(L_growth - 3.5) 6.052e-02
## h(L_usstock2000_adj - 6.53771):h(-3.5 - L_growth) 7.856e-02
## h(L_usstock2000_adj - 6.53771):h(L_lpercap - 10.2256) -9.912e+00
## h(L_usstock2000_adj - 6.53771):h(0.814617 - L_s_us) -1.885e-01
## h(L_usstock2000_adj - 6.53771):h(8.39968 - L_lnustrade) 5.019e-01
## h(L_hse.usnum - 20):h(-9 - L_polity2) -3.030e-01
## h(L_usstock2000_adj - 6.53771):h(10.2256 - L_lpercap):h(9.40286 - L_lnustrade) -1.324e-01
## h(L_usstock2000_adj - 6.53771):h(L_lpercap - 16.2256):h(L_lnustrade) 9.339e-01
## h(L_usstock2000_adj - 6.53771):h(L_lnustrade - 8.39968):h(21612.3 - allfdi2000) 2.408e-06
## h(L_usstock2000_adj - 6.53771):h(L_lnustrade - 8.39968):h(1.89 - L_growth) 3.078e-02
```

111 / 117

Notes

Type notes here...

112 / 117

Make PDP

```
seq_range <- function(x, n=25){  
  x <- na.omit(x);  
  seq(min(x), max(x), length=n)  
}  
usnum.s <- seq_range(bk$hse_usnum)  
predfun <- function(x,y){  
  z <- bk[,x];  
  z$L_hse_usnum <- y  
  predict(mod1, newdata=z)  
}  
res <- outer(test.samp, usnum.s, predfun)  
res <- t(apply(res, 1, function(x)x-mean(x)))  
k <- lapply(2:10, function(k)kmeans(res, centers=k))  
sapply(k, function(x)x$betweenss/(x$tot.withinss*x$betweenss))
```

```
## [1] 0.4453257 0.9076453 0.9475397 0.9785455 0.9440876 0.9890874 0.9893255  
## [8] 0.9899974 0.9941667
```

```
res2 <- rbind(k[[4]]$centers, res)  
d2 <- dist(res2)  
d2 <- as.matrix(d2)  
diag(d2) <- max(c(d2))  
closest <- apply(d2[1:5, ], 1, which.min)
```

113 / 117

```
library(purrr)  
tmp <- bk[test.samp[closest], ]  
tmp$cluster <- 1:5  
data <- map(1:5, ~as.list(tmp[, ])) %>%  
  map(~, -modify_at(.x, "L_hse_usnum", -usnum.s)) %>%  
  map(~, -do.call(data.frame, .x)) %>%  
  bind_rows()  
  
fits <- predict(mod1, newdata=data, se.fit=TRUE)  
data %>%  
  mutate(fit = fits$fit) %>%  
  group_by(cluster) %>%  
  mutate(fit = fit-mean(fit)) %>%  
  ungroup %>%  
  mutate(lwr = fit - 1.96*fits$se.fit,  
        upr = fit + 1.96*fits$se.fit)
```

Notes

Type notes here...

114 / 117

PDP

Cluster 1 2 3 4 5



115 / 117

Notes

Type notes here...

116 / 117

Significant Differences

For how many observations are there significant differences moving across the values of `L_hse_usnum`?

```
sigdiffs <- NULL
combs <- combn(25, 2)
D <- matrix(0, ncol=ncol(combs), nrow=25)
D[cbind(combs[1,], 1:ncol(combs))] <- -1
D[cbind(combs[2,], 1:ncol(combs))] <- 1
for(i in 1:length(test.samp)){
  dk <- as.list(bk[test.samp[i], ])
  dk$L_hse_usnum <- i
  A <- model.matrix(formula(lmod),
    data=do.call(data.frame, dk))
  pred <- A %>% coef(lmod)
  vpreds <- A %>% vcov(lmod) %>% t(A)
  diff <- c(t(D) %>% pred)
  vdiffs <- t(D) %>% vpreds %>% D
  tdiffs <- abs(diff)/sqrt(diag(vdiffs))
  tdiffs <- ifelse(is.finite(tdiffs), tdiffs, 0)
  sigdiffs <- c(sigdiffs, sum(tdiffs > 1.96))
}
```