STAT 705 Generalized additive models

Yen-Yi Ho

Department of Statistics, University of South Carolina

Stat 705: Data Analysis II

Generalized additive model (GAM)

GAMs were originally invented by Hastie and Tibshirani in 1986 (1, 2). GAMs relax the restriction that the relationship must be a simple weighted sum, and instead assume that the outcome can be modelled by a sum of arbitrary functions of each covariate.

1. Hastie, Trevor and Tibshirani, Robert. (1990), Generalized Additive Models, New York; Chapman and Hall.
2. Hastie, Trevor and Tibshirani, Robert. (1986), Generalized Additive Models, Statistical Science, Vol. 1, No 3, 297-318.

Generalized additive model

We have $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $y_1, \dots, y_n$ are normal, Bernoulli, or Poisson. The generalized additive model (GAM) is given by

$$h\{E(Y_i)\} = \beta_0 + g_1(x_{i1}) + \cdots + g_k(x_{ik}),$$

for p predictor variables. Y_i is a member of an exponential family such as binomial, Poisson, normal, etc. h is a link function.

Each of $g_1(x), \dots, g_p(x)$ are modeled via cubic smoothing splines, each with their own smoothness parameters $\lambda_1, \dots, \lambda_p$ either specified as $df_1, \dots, df_p$ or estimated through cross-validation. The model can be fit iteratively.

Generalized additive model

One example of this is through a basis expansion; for the jth predictor the transformation is:

$$g_j(x) = \sum_{k=1}^{K_j} \theta_{jk} \psi_{jk}(x),$$

where $\{\psi_{jk}(\cdot)\}_{k=1}^{K_j}$ are B-spline basis functions, or sines/cosines, etc. This approach has gained more favor from Bayesians. Cubic smoothing splines is also a popular choice.
vspace0.2in
This is an example of "nonparametric regression," which ironically connotes the inclusion of *lots* of parameters rather than fewer.

Choosing λ

Hastie and Tibshirani (1986, 1990) point out that the meaning of λ depends on the units x_i is measured in, but that λ can be picked to yield an "effective degrees of freedom" df or an "effective number of parameters" being used in $g(x)$. Then the complexity of $g(x)$ is equivalent to $(df - 1)$-degree polynomial, but with the coefficients "spread out" more yielding a more flexible function that fits data better.

λ can be picked through cross validation, by minimizing

$$CV(\lambda) = \sum_{i=1}^{n} (y_i - g_\lambda^{-i}(x_i))^2.$$

Estimation: local scoring algorithm

$$
\begin{aligned}
E(Y|X) &= \mu \\
h(\mu) &= \eta(X) \\
\eta &= \beta_0 + \sum_i^p g_i(X_i)
\end{aligned}
$$

Estimate $g_i(\cdot)$ through backfitting algorithm. For example, for a simple covariate Gaussian Y

- Initialization: $\beta_0 = E(Y), s_1^1(\cdot) = ... = s_p^1(\cdot) = 0, m = 0$
- Define $R_j = Y - \beta_0 - \sum_{k=1}^{j-1} - \sum_{k=j+1}^{p}$, then fit $s_j^m = E(R_j|X_j)$.
- Until $RSS = E[Y - \beta_0 - \sum_{k=1}^{p}]^2$ fail to decrease

Estimation: local likelihood

The main idea behind local likelihood is to locally fit parametric models by maximum likelihood. For linear regression as an example:

$$l_{x,h}(\beta) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}[Y_i - \beta_0 - \beta_1(X_i - x)\ldots - \beta_p(X_i - x)]^2 K_h(x - X_i).$$

Minimize the local likelihood w.r.t β.

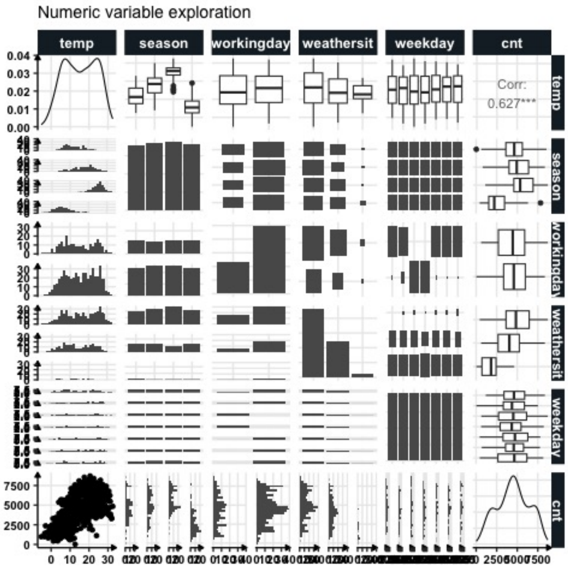
Example: bike share data

```
> url<-"https://people.stat.sc.edu/hoyen/STAT705/Data/bike.csv"
> bikes<-read.csv(url)
> str(bikes)
'data.frame': 731 obs. of  12 variables:
 $ season         : chr  "WINTER" "WINTER" "WINTER" "WINTER" ...
 $ yr             : int  2011 2011 2011 2011 2011 2011 2011 2011 2011 2011 ...
 $ mnth           : chr  "JAN" "JAN" "JAN" "JAN" ...
 $ holiday        : chr  "NO HOLIDAY" "NO HOLIDAY" "NO HOLIDAY" "NO HOLIDAY" ...
 $ weekday        : chr  "SAT" "SUN" "MON" "TUE" ...
 $ workingday     : chr  "NO WORKING DAY" "NO WORKING DAY" "WORKING DAY" "WORKING DAY" ...
 $ weathersit     : chr  "MISTY" "MISTY" "GOOD" "GOOD" ...
 $ temp           : num  8.18 9.08 1.23 1.4 2.67 ...
 $ hum            : num  80.6 69.6 43.7 59 43.7 ...
 $ windspeed      : num  10.7 16.7 16.6 10.7 12.5 ...
 $ cnt            : int  985 801 1349 1562 1600 1606 1510 959 822 1321 ...
 $ days_since_2011: int  0 1 2 3 4 5 6 7 8 9 ...
```

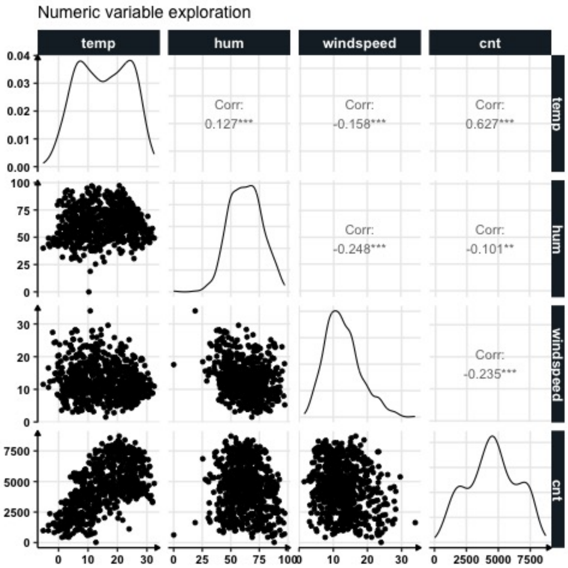
Example: bike share data

```
> head(bikes)
  season    yr mnth    holiday weekday     workingday weathersit     temp     hum
1 WINTER 2011  JAN NO HOLIDAY     SAT NO WORKING DAY      MISTY 8.175849 80.5833
2 WINTER 2011  JAN NO HOLIDAY     SUN NO WORKING DAY      MISTY 9.083466 69.6087
3 WINTER 2011  JAN NO HOLIDAY     MON    WORKING DAY       GOOD 1.229108 43.7273
4 WINTER 2011  JAN NO HOLIDAY     TUE    WORKING DAY       GOOD 1.400000 59.0435
5 WINTER 2011  JAN NO HOLIDAY     WED    WORKING DAY       GOOD 2.666979 43.6957
6 WINTER 2011  JAN NO HOLIDAY     THU    WORKING DAY       GOOD 1.604356 51.8261
  windspeed  cnt days_since_2011
1 10.749882  985               0
2 16.652113  801               1
3 16.636703 1349               2
4 10.739832 1562               3
5 12.522300 1600               4
6  6.000868 1606               5
```

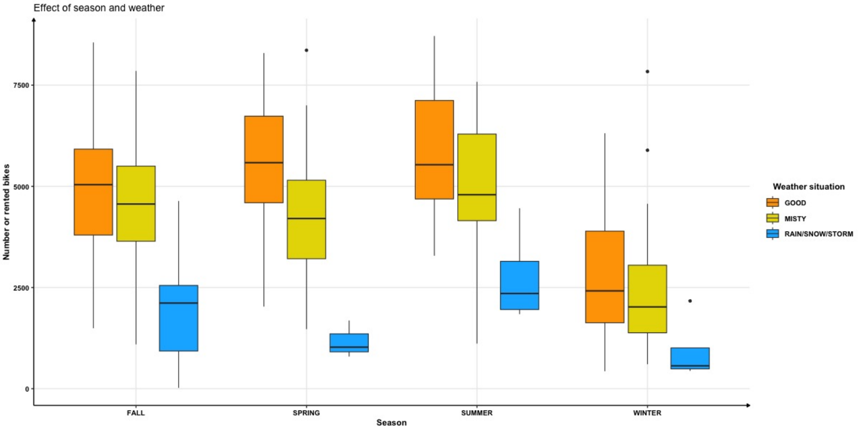
Explore the data



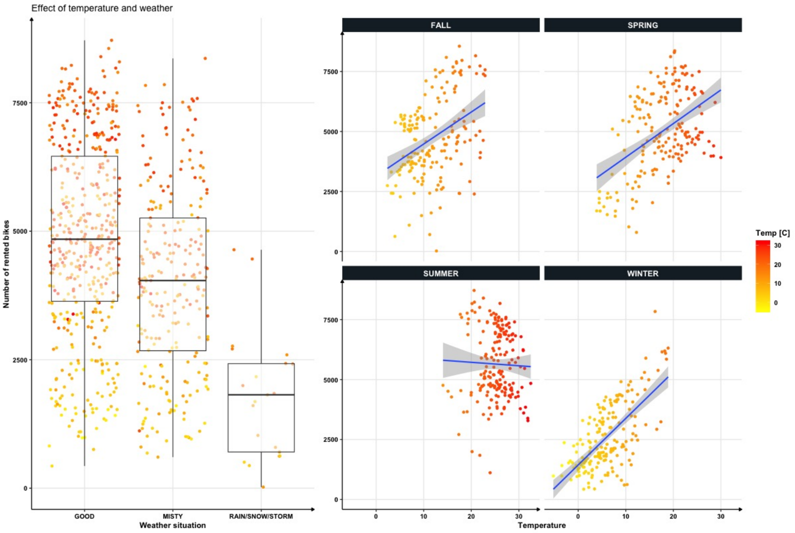
Explore the data



Temperature and weather



Temperature and weather



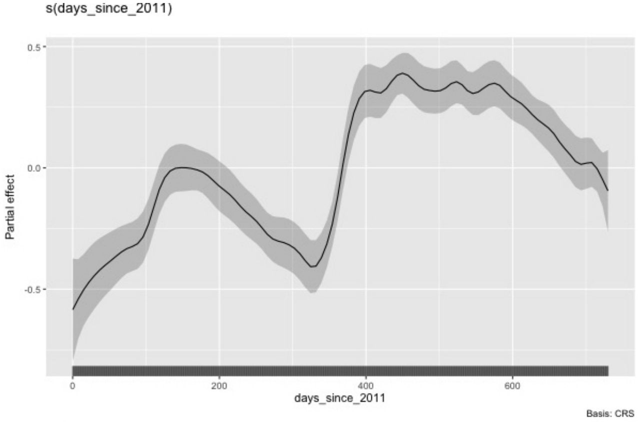
GAM model

```
> library(mgcv)
> M2 = gam(cnt ~ season + weathersit  + s(days_since_2011, bs ="cr", k = 70) +
+           s(temp, bs = "cr", by = season, k = 15), data = bikes, family=quasipoisson(link = "log"))
>
> summary(M2)
Family: quasipoisson
Link function: log
Formula:
cnt ~ season + weathersit + s(days_since_2011, bs = "cr", k = 70) +
    s(temp, bs = "cr", by = season, k = 15)

Parametric coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                8.67573    0.06583 131.781  < 2e-16 ***
seasonSPRING              -0.36329    0.08615  -4.217 2.81e-05 ***
seasonSUMMER               0.11888    0.11224   1.059     0.29
seasonWINTER              -0.39112    0.08577  -4.560 6.05e-06 ***
weathersitMISTY           -0.15401    0.01337 -11.521  < 2e-16 ***
weathersitRAIN/SNOW/STORM -0.87218    0.05563 -15.677  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
                        edf Ref.df       F  p-value
s(days_since_2011)   25.280 31.496  48.287  < 2e-16 ***
s(temp):seasonFALL    5.035  6.167   9.995  < 2e-16 ***
s(temp):seasonSPRING  2.751  3.487  14.882  < 2e-16 ***
s(temp):seasonSUMMER  2.098  2.647  18.589 7.19e-07 ***
s(temp):seasonWINTER  1.000  1.001 104.113  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =   0.88   Deviance explained = 87.5%
GCV = 128.74  Scale est. = 109.35    n = 731
```

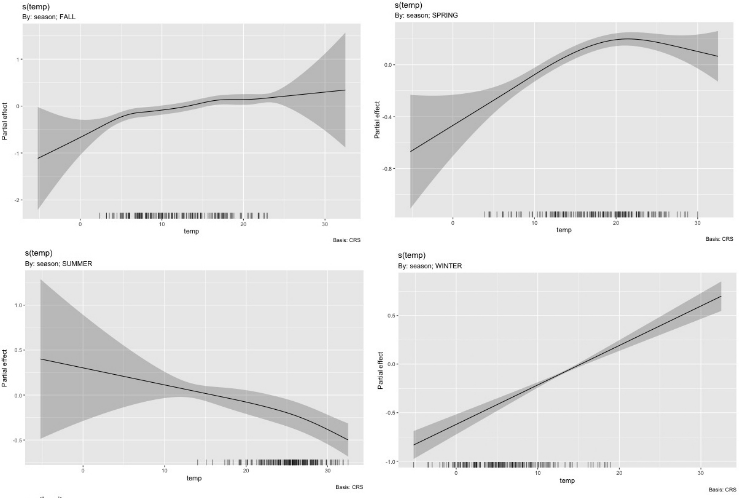
Checking K value and edf

```
> k.check(M2)
                         k'       edf   k-index p-value
s(days_since_2011)    69 25.280111 0.7871581  0.0000
s(temp):seasonFALL    14  5.034919 0.9175327  0.0075
s(temp):seasonSPRING 14  2.751155 0.9175327  0.0250
s(temp):seasonSUMMER 14  2.097587 0.9175327  0.0100
s(temp):seasonWINTER 14  1.000275 0.9175327  0.0100
```

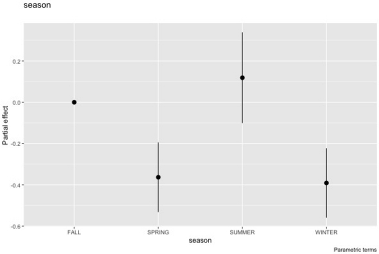
Results


s(days_since_2011)
0.5
0.0
-0.5
Partial effect
0
200
400
600
days_since_2011
Basis: CRS

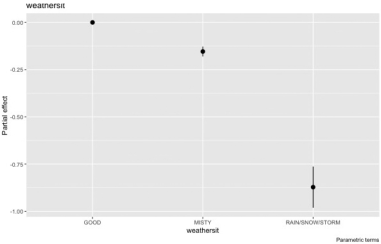
Results


s(temp)
By: season; FALL
Partial effect
temp
Basis: CRS
s(temp)
By: season; SPRING
Partial effect
temp
Basis: CRS
s(temp)
By: season; SUMMER
Partial effect
temp
Basis: CRS
s(temp)
By: season; WINTER
Partial effect
temp
Basis: CRS

Results



season
0.2
0.0
-0.2
-0.4
-0.6
Partial effect
FALL
SPRING
SUMMER
WINTER
season
Parametric terms



weathersit
0.00
-0.25
-0.50
-0.75
-1.00
Partial effect
GOOD
MISTY
RAIN/SNOW/STORM
weathersit
Parametric terms