STA314, Lecture 8

November 8, 2019
Stanislav Volgushev

The prediction competition

This weekend, competition will be set up on kaggle.

- 15% of the final grade will be homework assignments related to R and competition and due within the next two weeks (several smaller assignments with different due dates). **No team work is allowed for this part**.
- **Teams of up to 3** will be allowed for competition.
- Team can gain up to 10% *extra credit* (10% for top team, 9% for second etc, top 10 teams get extra credit). Amount will depend on team performance in the competition. **Extra credit will be split equally between team members, rounded down.**
- **If your team ranks highly, you will need to explain your approach to instructor in dedicated office hours and provide R code. If your explanation is not satisfactory you will get 0 extra credit.**
- **The competition will end on Nov 28. Top ranked performers will present their approach in lectures on Nov 29 (required for extra credit).**

The prediction competition: technical details

- The competition will be using *kaggle in class*. You will need to register a free account with the *kaggle* website.

- You will be able to download a training set (predictors and responses) and the predictors in a test set from the website. Train your model on the training set. Predictions for the test set can be submitted to the webpage.

- You can submit predictions once per day (once every 24 hr) to see how well you are doing.

- Submitted predictions will be evaluated on a random part of the test set, this will lead to the public leader board. Final results are computed on the whole test set, this will be the final competition results.

Comment on midterm question

Question: Consider a model of the form $y_i = f(x_i) + \varepsilon_i$, where ε_i are i.i.d. with $E[\varepsilon_i] = 0, \mathrm{Var}\,(\varepsilon_i) = 2$ and assume that $\widehat{f}(x_0)$ is an estimator that you obtained on the training set. Consider a new observation $y_0 = f(x_0) + \varepsilon_0$ where x_0 is fixed and ε_0 is independent of $\widehat{f}(x_0)$ and satisfies $E[\varepsilon_0] = 0, \mathrm{Var}\,(\varepsilon_0) = 2$. Is it possible that $E[(\widehat{f}(x_0) - f(x_0))^2] = 1$? Justify your answer to get full marks.

Answer: yes, this is possible! See explanation on blackboard for more details.

Note that $E[(\widehat{f}(x_0) - f(x_0))^2]$ does not include the irreducible part of the prediction MSE! The answer would have been **no, not possible** if the question was about $E[(\widehat{f}(x_0) - Y_0)^2]$.

Recap: splines

Towards the end of previous lecture we discussed *polynomial splines*.

Polynomial spline of degree d with knots $c_1, ..., c_K$:

- polynomial of degree d on intervals $(-\infty, c_1], (c_2, c_2], ..., (c_{K-1}, c_K], (c_k, +\infty)$.
- $d - 1$ times continuously differentiable on $\mathbb{R}$.

Pro: gives smooth functions $\hat{f}$, can describe complicated functions f. Con: need to pick knots which can influence performance a lot and is difficult to do in automatic fashion.

Next: methods that avoid selection of knots.

Smoothing splines

Smoothing splines

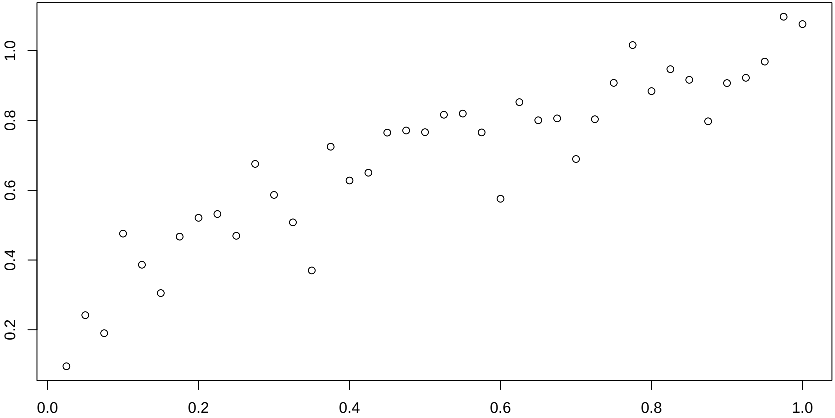
Smoothing splines avoid need to choose location of knots manually. Flexibility of smoothing splines is controlled by a 'penalty parameter', similar to ridge and lasso.

Given data $(x_i, y_i)_{i=1,\ldots,n}$, the corresponding smoothing spline $\hat{f}$ is defined as

$$\hat{f} := \arg\min_{g \in \mathcal{C}^2} \left\{ \sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int (g''(x))^2 dx \right\}$$

- $\mathcal{C}^2$ denotes the space of all functions $\mathbb{R} \to \mathbb{R}$ which have two continuous derivatives (smooth functions).
- $g''(x)$ is the second derivative of g in point x.
- The first part $\sum_{i=1}^{n}(y_i - g(x_i))^2$ corresponds to the training MSE when predicting y_i with $g(x_i)$. This is similar as RSS for linear models.
- Question 1: why not just minimize $\sum_{i=1}^{n}(y_i - g(x_i))^2$? What will happen?

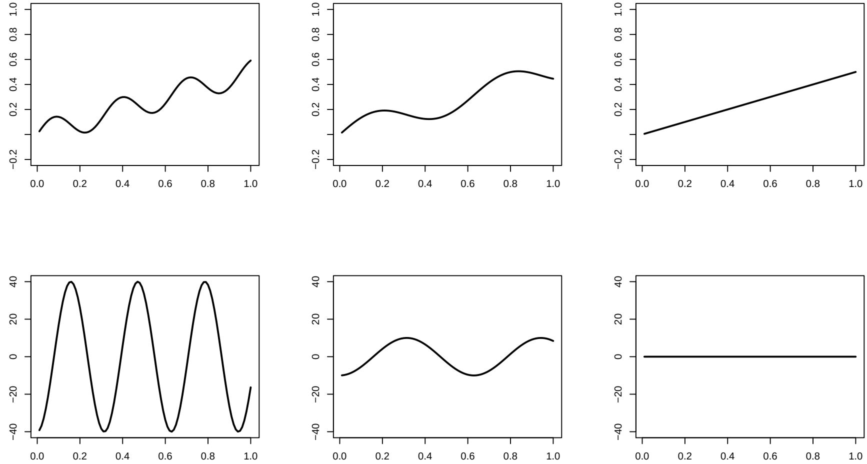
Why not just minimize $\sum_{i=1}^{n}(y_i - g(x_i))^2$?



Why not just minimize $\sum_{i=1}^{n}(y_i - g(x_i))^2$?

What is the effect of $\int (g''(x))^2 dx$?

Plots of g (top row) and g'' (bottom row) for some examples of functions g (scale on y axis same for all 3 plots in bottom row). More 'wiggles' in g mean larger g''.



- Left to right: $\int (g''(x))^2 dx \approx 814,\ 52,\ 0$.
- Adding $\lambda \int (g''(x))^2 dx$ to $\sum_{i=1}^{n} (y_i - g(x_i))^2$ will prevent g from overfitting.

Smoothing splines: continued

Theorem For any $\lambda > 0$ the function $\hat{f}$ defined by

$$\hat{f} := \arg\min_{g \in \mathcal{C}^2} \sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int (g''(x))^2 dx$$

is *natural cubic spline with knots at* $x_1, ..., x_n$!

Hence $\hat{f}$ can be represented in the form

$$\hat{f}(x) = \sum_{k=1}^{n} \hat{b}_k g_k(x)$$

for some basis functions $g_1, ..., g_n$ (which depend on data, we omit the details). For computing $\hat{f}$: we just need to maximize over a finite number of parameters!

Advantage of smoothing splines over basis expansion: in smoothing splines, we only need to select the parameter λ (positive number), the location and number of knots is selected automatically! The parameter λ can be selected by cross-validation.

- There is a way to specify 'degrees of freedom' for smoothing splines, we omit the details here (ask me after lectures if you are interested).

Local polynomial regression

Local polynomial regression

Degree d, *kernel* $K : [0,1] \to \mathbb{R}$, *span* $s \in (0,1)$ computed at x_0.

---

1. Let k be integer part of ns. Select k points among $x_1, \ldots, x_n$ that are closest to x_0, denote those by $z_1, \ldots, z_k$, and denote corresponding response values by $y_1^*, \ldots, y_k^*$. Let $m := \max\{|x_0 - z_1|, \ldots, |x_0 - z_k|\}$
2. Consider the minimization problem

$$\hat{B} := \arg \min_{(b_1, \ldots, b_{d+1})^\top \in \mathbb{R}^{d+1}} \sum_{i=1}^{k} K(|x_0 - z_i|/m)(y_i^* - b_1 - b_2 z_i - \ldots - b_{d+1} z_i^d)^2$$

3. Regression function at x_0 is of the form $\hat{f}(x_0) = \hat{b}_1 + x_0 \hat{b}_2 + \ldots + \hat{b}_{d+1} z_i^d$.

Special case: $d = 0$ (local constant regression). Let $W_i(x_0) := K(|x_0 - z_i|/m)$. Then (see blackboard for derivation)

$$\hat{f}(x_0) = \hat{b}_1 = \frac{\sum_{i=1}^k W_i(x_0) y_i^*}{\sum_{i=1}^k W_i(x_0)},$$

i.e. *weighted average* of y_i corresponding to k nearest points.

Local constant regression: intuition

Special case: $d = 0$ (local constant regression). Let $W_i(x_0) := K(|x_0 - z_i|/m)$. Then (see blackboard for derivation)

$$\hat{f}(x_0) = \hat{b}_1 = \frac{\sum_{i=1}^k W_i(x_0) y_i^*}{\sum_{i=1}^k W_i(x_0)},$$

i.e. *weighted average* of y_i corresponding to k nearest points.

Assume that $y_i = f(x_i) + \varepsilon_i$, $\mathbb{E}[\varepsilon_i] = 0$, $Var(\varepsilon_i) = \sigma^2$. Then

$$\mathbb{E}[\hat{f}(x_0)] - f(x_0) = \frac{\sum_{i=1}^k W_i(x_0) f(z_i)}{\sum_{i=1}^k W_i(x_0)} - f(x_0) = \sum_{i=1}^k \frac{W_i(x_0)}{\sum_{\ell=1}^k W_\ell(x_0)} (f(z_i) - f(x_0))$$

$$Var(\hat{f}(x_0)) = \sigma^2 \sum_{i=1}^k \Big(\frac{W_i(x_0)}{\sum_{\ell=1}^k W_\ell(x_0)} \Big)^2$$

- Can try to balance bias and variance by choice of kernel K.
- Since $f(z_i) - f(x_0)$ larger if $|z_i - x_0|$ larger give less weight to such observations.
- Intuition for local polynomial regression of arbitrary orders similar.

Local polynomial regression: motivation

What can we gain by considering local polynomial regression with degree > 0?

Boundary effects: see picture on blackboard and simulation

General motivation: Taylor expansion. If a function f is d times continuously differentiable at x_0, then by Taylor expansion

$$f(z) = f(x_0) + f^{(1)}(x_0)(z - x_0) + ... + \frac{1}{d!} f^{(d)}(x_0)(z - x_0)^d + r(x_0, z)$$

where $|r(x_0, z)|/|x_0 - z|^d \to 0$ for $z \to x_0$.

- The blue part is a polynomial in z.
- The red part is a remainder, 'small' if $|x_0 - z|$ small.
- Local polynomial regression tries to model the blue part.

Local polynomial regression: closing comments

- For proper choice of kernel K (i.e. $K(1) = 0$, K continuous) this leads to functions $\hat{f}$ which are continuous. Advantage compared to k-nn.

- Typical choice of degree d is $d = 1$, sometimes $d = 3$.

- There are many variations of 'local weighting idea', for example using all points in a local window of given length.

- One of many **R** implementations: `loess` function in `splines` package. Uses kernel $K(x) = (1 - x^3)^3$.

- Replacing $|x_0 - z_i|$ by other way of measuring distances (e.g. Euklidean distance), this can be extended to predictors with values in $\mathbb{R}^p$. Typically doe snot work well beyond about $p = 3$. This is because of the 'curse of dimensionality'.

Some R examples: file `Sta314-09-Lecture07-SmoothingSplinesLocalregr.R`

Generalized additive models (GAM)

Generalized Additive Models (GAM): Motivation

Methods we have discussed so far:

1. Linear models with several predictors. Have discussed ways to select relevant predictors in a lot of details.
2. Methods that allow one predictor to have a non-linear impact: k-nn, basis expansions (step functions, piecewise polynomials, local polynomial regression), smoothing splines.
3. Methods that allow several predictors to simultaneously have non-linear impact: k-nn and local regression.

As mentioned in previous lectures, k-nn and local regression can be problematic if predictors have 'high' dimension (≥ 4). Reason: too much variance.

We also have seen that including just impact of one predictor does not perform well. Reason: too much bias.

This lecture: methods that allow for non-linear impact of several predictors but still put some restrictions on overall impact. **G**eneralized **A**dditive **M**odels.

Generalized Additive Models

Generalized additive models try to find a balance between fully non-parametric model specifications and simple linear models. Setting: predictors $X_i = (x_{i,1}, ..., x_{i,d})^\top \in \mathbb{R}^d$, $y_i = f(X_i) + \varepsilon_i$.

Additive model for f:

$$f(X_i) = g_1(x_{i,1}) + g_2(x_{i,2}) + ... + g_d(x_{i,d}).$$

- The functions $g_1, ..., g_d$ can be non-linear.
- Each function depends on just one predictor, so this puts a lot of structure on the function f. The structure is *additive*.
- Aim of imposing linear structure: reduce variance.
- Function g_k describes effect of $x_{i,k}$ while holding others fixed.

Example: $X = (\texttt{TV},\texttt{Radio},\text{News})$ which of the following functions can be written in above form?

- f(X) $= b_1 + b_2\exp(\texttt{TV}) + b_3\exp(\texttt{Radio}) + b_4\texttt{News}^2$
- f(X) $= \exp(b_1\texttt{TV} + b_2\texttt{Radio})$
- f(X) $= b_1 + b_2\texttt{TV} + b_3\texttt{TV}^2 + b_4\exp(\texttt{Radio}) + b_5\texttt{Radio} + b_6\texttt{News}^3$

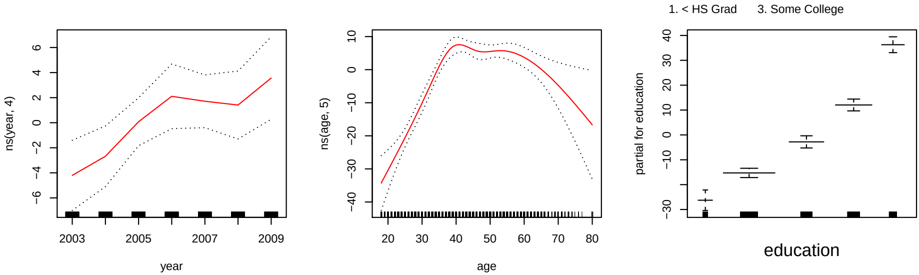
Fitting a GAM with `lm()`

One approach to such fitting GAM's in R: use basis expansions we discussed during previous lectures. Examples: polynomial splines, natural cubic splines etc.

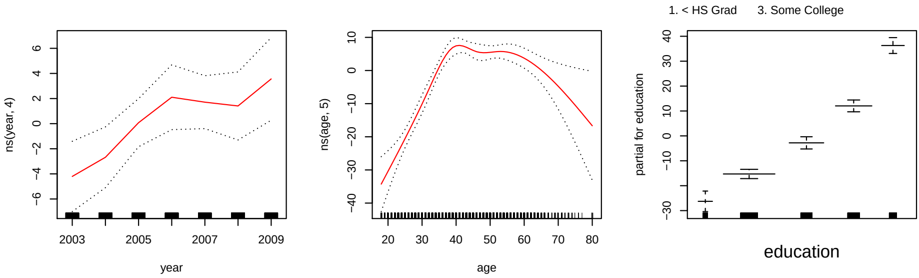
Example 1: **Wage** data set. Use predictors `age`, `year`, `education` to predict `wage`. Note: `education` is qualitative. Use model specification

```
gam1 = lm(wage~ns(year,4)+ns(age,5)+education,data=Wage)
```

Nice way to plot this: `plot.gam` function, see R examples.



GAM for wage



- Dotted lines on left two graphs, bars on right graph: confidence intervals for effect of predictor at this predictor value.
- Holding `education` and `age` fixed, slight increase with `year`. Perhaps due to inflation.
- Holding `age` and `year` fixed: more education corresponds to higher salary. Good news for all of you! Additional motivation to move from middle (some college) to fourth (college graduate).

Fitting a GAM with 'backfitting' via the gam package in R

We have seen other approaches to modelling non-linear impact of predictors such as smoothing splines and local polynomial regression that can not be handled by the lm() function. Some of those methods can be applied in GAM's using the gam package.

Example: **Wage**, continued. Previously, we were using natural cubic splines. Now use local regression with span 0.7 for the effect of age and a smoothing spline with 4 degrees of freedom for the effect of year

```
gam.m3=gam(wage~s(year,df=4)+lo(age,span=0.7)+education,data=Wage)
```

- s(year,4) indicates that the effect of year should be modelled via a smoothing spline with 4 degrees of freedom
- lo(age,span=0.7) indicates that effect of age should be modelled using local regression with span 0.7
- education is kept as dummy variable.
- lo and s can be applied to arbitrary combinations of predictors, and applied together with other specifications compatible with lm().

The summary function for GAM

```
gam.m3=gam(wage~s(year,df=4)+lo(age,span=0.7)+education,data=Wage)
summary(gam.m3)
```

Some of the output

```
Anova for Parametric Effects
                        Df  Sum Sq Mean Sq F value    Pr(>F)
s(year, df = 4)        1.0   25188   25188  20.255 7.037e-06 ***
lo(age, span = 0.7)    1.0  195537  195537 157.243 < 2.2e-16 ***
education              4.0 1101825  275456 221.511 < 2.2e-16 ***
Residuals           2988.8 3716672    1244
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

Which predictors have *linear* effect in the presence of other predictors? Note: this is for just *Parametric Effect*, indicates if it makes sense to include predictor in the model (from a prediction point of view, similar idea to linear models when comparing RSS).

The summary function for GAM II

```
gam.m3=gam(wage~s(year,df=4)+lo(age,span=0.7)+education,data=Wage)
summary(gam.m3)
```

Some of the output

```
Anova for Nonparametric Effects
                    Npar Df Npar F  Pr(F)
(Intercept)
s(year, df = 4)         3.0  1.103 0.3464
lo(age, span = 0.7)     1.2 88.835 <2e-16 ***
education
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Which predictors have *Nonparametric Effect* effect in the presence of other predictors? *Nonparametric Effect* means that we should include a *non-linear* transformation of the predictor (from a prediction point of view)? Note: `Npar Df` corresponds to 'non-parametric degrees of freedom', i.e. something measuring 'number of parameters'.

Comparing GAM models via ANOVA

The previous results indicate that it does not make sense to include `year` as a non-parametric predictor. An alternative way to look at this is via the function `anova()` applied to the output of `gam()`. Setup: order candidate models in order of increasing complexity and compare, each to the next.

```
  gam.m1=gam(wage~ s(age,df=5)+education,data=Wage)
  gam.m2=gam(wage~ year +s(age,df=5)+education,data=Wage)
  gam.m3=gam(wage~s(year,df=4)+s(age,df=5)+education,data=Wage)
  anova(gam.m1,gam.m2,gam.m3)
```

Will give output

```
Model 1: wage ~ s(age, df = 5) + education
Model 2: wage ~ year + s(age, df = 5) + education
Model 3: wage ~ s(year, df = 4) + s(age, df = 5) + education
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      2990    3711731
2      2989    3693842  1  17889.2 0.0001419 ***
3      2986    3689770  3   4071.1 0.3483897
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comparing GAM models via ANOVA II

```
Model 1: wage ~ s(age, df = 5) + education
Model 2: wage ~ year + s(age, df = 5) + education
Model 3: wage ~ s(year, df = 4) + s(age, df = 5) + education
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      2990    3711731
2      2989    3693842  1  17889.2 0.0001419 ***
3      2986    3689770  3   4071.1 0.3483897
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

- this is sequentially testing if model 2 is significantly better compared to model 1 and if model 3 is better compared to model 2.
- output indicates that it makes sense to include `year`, but including `year` in a non-linear fashion does not help much.
- `Resid.` `Dev` is simply sum of squared residuals (for this particular way of running GLM). `Deviance` is reduction in RSS.

Extending GAM for better prediction: adding interactions

The idea behind GAM is to impose an additive structure in order to control the variance. However, a completely additive structure is not always adequate. Recall the example of predicting `Sales` from `TV`, `Radio`, `Newspaper`. To achieve the best performance we needed *interactions*.

Adding interactions to GAM in **R** is easy since interactions preserve linear model structure. Example: model for predicting `Sales` from `TV` and `Radio`

```
  gam(Sales ~ Radio + TV:Radio + s(TV,df=5))
```

For data sets with many predictors, capturing the important interactions can significantly improve prediction performance.

Generalizing GAM further

The idea behind GAM is to impose an additive structure in order to control the variance. However, a completely additive structure is not always adequate. One generalization discussed on the previous slide is to add interactions. An even more general version is to allow some predictors to enter jointly.

Example: assume we consider a model of the form

$$f(X_i) = g_1(x_{i,1}, x_{i,2}) + g_2(x_{i,3})$$

Unless $g_1(x_{i,1}, x_{i,2}) = h_1(x_{i,1}) + h_2(x_{i,2})$ this is not a GAM. Still, this is less general than allowing for arbitrary functions f and can be helpful to achieve a good bias-variance trade-off.

Example: model for predicting `wage` from `age`, `year` and `education` with general function $g_1(age, year)$

```
gam.np=gam(wage~lo(age,year,span=0.7)+education,data=Wage)
```

Whether this generalization helps to make better predictions depends on the data generating mechanism...

Some R examples: file `Sta314-Lecture08-GAM.R`