

# Statistical Computing

Michael Mayer

March 2023



# Statistical Computing: What will we do?

## Chapters

1. R in Action
2. Statistical Inference
3. Linear Models
4. Model Selection and Validation
5. Trees
6. Neural Nets

## Remarks

- ▶ Chapters 3 to 6:  
Statistical ML in Action
- ▶ Two weeks per chapter
- ▶ Exercises at end of chapter notes

# Linear Models

# Outline

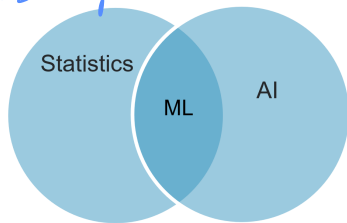
- ▶ Start of “Statistical ML in Action”
- ▶ Linear Regression
- ▶ Generalized Linear Models (GLM)
- ▶ Modeling Large Data

# Statistical ML in Action

## What is ML?

Collection of statistical algorithms used to

1. predict things (supervised ML) or to
2. investigate data structure (unsupervised)
  - *principal component analysis*
  - *clustering*



## Focus on supervised ML

- ▶ Regression *numerical*
- ▶ Classification *categorical*

## Chapters

3. Linear Models
4. Model Selection and Validation
5. Trees
6. Neural Nets

## Model Setup

*usually the expectation  
but can be median, quantile, VaR*

$$T(Y \mid \mathbf{X} = \mathbf{x}) \approx f(\mathbf{x})$$

This means: Approximate property  $T$  of **response**  $Y$  (often  $T = \mathbb{E}$ ) by function  $f$  of  $p$ -dim **covariate** vector  $\mathbf{X} = (X^{(1)}, \dots, X^{(p)})$  with value  $\mathbf{x} = (x^{(1)}, \dots, x^{(p)})$

- ▶ Estimate  $f$  by  $\hat{f}$  from data by minimizing objective

$$Q(f) = \sum_{i=1}^n L(y_i, f(\mathbf{x}_i)) + \lambda \Omega(f)$$

- ▶  $L$ : loss function in line with  $T$ , e.g. squared error  $L(y, z) = (y - z)^2$  for  $T = \mathbb{E}$
- ▶  $\lambda \Omega(f)$ : optional penalty *like forecasting*
- ▶  $\mathbf{y} = (y_1, \dots, y_n)^T$ : observed values of  $Y$
- ▶  $\mathbf{x}_1, \dots, \mathbf{x}_n$ :  $n$  feature vectors;  $x_i^{(j)}$ :  $i$ -th value of  $X^{(j)}$ ;  $\mathbf{x}^{(j)}$ :  $n$  values of feature  $X^{(j)}$

# Linear Regression

- ▶ Postulate model equation *assume*

$$\mathbb{E}(Y | \mathbf{x}) = f(\mathbf{x}) = \beta_0 + \beta_1 x^{(1)} + \dots + \beta_p x^{(p)}$$

- ▶ Interpretation of parameters  $\beta_j$ ? Ceteris Paribus!
- ▶ Optimal  $\hat{\beta}_j$ ? Minimize as objective the sum of squared errors/residuals

$$\sum_{i=1}^n \underbrace{(y_i - \hat{y}_i)}_{\text{Residual}}^2$$

*fitted value*

- ▶ Predicted/fitted values  $\hat{y}_i = \hat{f}(\mathbf{x}_i)$
- ▶ This means: we work with the squared error loss and no penalty

## Example

Simple linear regression:  $\mathbb{E}(Y | x) = \alpha + \beta x$

# Aspects of Model Quality

## Predictive performance *independent of the sample size*

▶  $\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$

*absolute measure*  
*relative measure*

▶ Root-MSE (RMSE) *sqr of MSE*

▶ Relative performance:

$$R^2 = 1 - \text{MSE} / \text{MSE}_0$$

▶  $\text{MSE}_0 \rightarrow$  intercept-only model

## Validity of assumptions

▶ Model equation is correct *main assumption*

▶ **Normal** linear model

$$Y = f(\mathbf{x}) + \varepsilon \text{ with } \varepsilon \sim N(0, \sigma^2)$$

*if assumptions are correct,  
statistical inference is exact*

## Example



## Typical Problems

very hard to deal  
with, need models  
to model missing  
values

### Missing values

in the covariates  
in the response

### Outliers

### Overfitting

more covariates  
than samples

### Collinearity

not such a problem,  
but the more dependent, the  
more difficult it gets to interpret  
the  $\beta$ 's

# Categorical Covariates

OHE just one column  
is "hot"

## Example of One-Hot-Encoding

- ▶ One-Hot-Encoding
- ▶ Dummy coding → has one column less  
always in comparison  
to one category
- ▶ Interpretation?

color	D	E	F	G	H	I	J
E	0	1	0	0	0	0	0
E	0	1	0	0	0	0	0
E	0	1	0	0	0	0	0
I	0	0	0	0	0	1	0
J	0	0	0	0	0	0	1
J	0	0	0	0	0	0	1
I	0	0	0	0	0	1	0
H	0	0	0	0	1	0	0
E	0	1	0	0	0	0	0
H	0	0	0	0	1	0	0

Example

# Linear Regression is Flexible

1. Non-linear terms
2. Interactions
3. Transformations like logarithms

often needed to make  
a good model

in practice  $\rightarrow$  random forest  
boosted models

These elements are essential but tricky!

$\rightarrow$  see the transformations  
in the ML models  
create a LM from them,  
almost as good as ML  
but interpretable

# Non-Linear Terms

## Deal with non-linear associations to $Y$ ?

→ invest more parameters

### 1. Polynomial terms

- ▶ E.g., cubic regression

$$\mathbb{E}(Y | x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

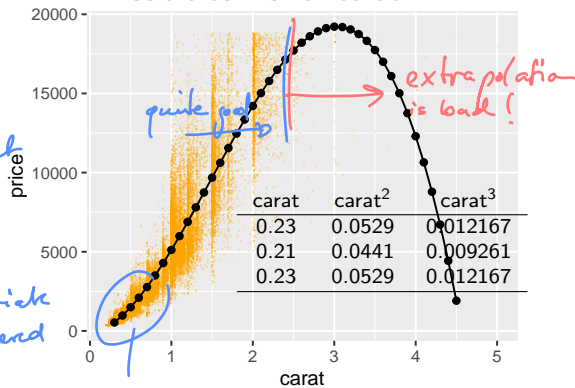
- ▶ Don't extrapolate!

### 2. Regression splines

→ relatively large data

multicollinearity  
don't interpret  $\beta$  directly  
~50 rows, every additional covariate must be considered

## Cubic terms for carat



Use systematic predictions

# Interactions

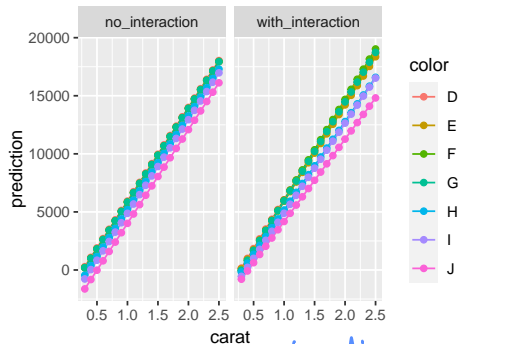
- ▶ Additivity of effects not always realistic

$$\mathbb{E}(Y \mid \mathbf{x}) = \beta_o + \beta_1 x^{(1)} + \dots + \beta_p x^{(p)}$$

- ▶ Adding interaction terms brings necessary flexibility  $\rightarrow$  more parameters
- ▶ Interaction between features  $X$  and  $Z$ 
  - ▶ Multiplication (for categoricals?)
  - ▶ For categorical  $Z$ , effects of  $X$  are calculated by level of  $Z$
  - ▶ Like separate models per level of  $Z$

interactions can lead to a lot of parameters fast

## Carat and color



interaction  
between color & carat

# Transformations of Covariates

## Examples

- ▶ Dummy variables for categoricals
- ▶ Decorrelation
- ▶ Logarithms against outliers

Effects are interpreted for transformed covariates

# Logarithmic Covariates

- ▶  $\mathbb{E}(Y | x) = \alpha + \beta \log(x)$
- ▶ Properties of logarithm allow interpretation **for original covariate**
- ▶ A 1% increase in  $X$  is associated with an increase in  $\mathbb{E}(Y)$  of about  $\beta/100$
- ▶ Why?

$$\begin{aligned}\mathbb{E}(Y | 1.01x) - \mathbb{E}(Y | x) &= \alpha + \beta \log(1.01x) - \alpha - \beta \log(x) \\ &= \beta \log\left(\frac{1.01x}{x}\right) \\ &= \beta \log(1.01) \approx \beta/100\end{aligned}$$

## Example

# Logarithmic Responses

We see: log-transforming  $X$  allows to talk about relative effects in  $X$

Idea: log-transformed  $(Y)$  allows to talk about relative effects on  $(Y)$

Assume for a moment that

$$\mathbb{E}(\log(Y) \mid x) = \alpha + \beta x \implies \log(\mathbb{E}(Y \mid x)) = \alpha + \beta x$$

*wrong, just for simplicity*

- ▶ Multiplicative model  $\mathbb{E}(Y \mid x) = e^{\alpha + \beta x}$
- ▶ Relative interpretation: "A one-point increase in  $X$  is associated with a relative increase in  $\mathbb{E}(Y)$  of  $100\%(e^{\beta} - 1) \approx 100\%\beta$ "
- ▶ If also  $\log(X)$ ?

But assumption is wrong  $\rightarrow$  biased predictions for  $Y \rightarrow$  GLMs

*bias introduced & can be a problem*

Examples



## Example: Realistic Model for Diamond Prices

- ▶ Response:  $\log(\text{price})$
- ▶ Covariates:  $\log(\text{carat})$ , color, cut and clarity



# Generalized Linear Model (GLM)

(One) extension of linear regression

## Model equation

Two equivalent formulations

$$g(\mathbb{E}(Y | \mathbf{x})) = \eta(\mathbf{x}) = \beta_0 + \beta_1 x^{(1)} + \dots + \beta_p x^{(p)}$$

$$\mathbb{E}(Y | \mathbf{x}) = \underbrace{g^{-1}}_{\text{mapping}}(\underbrace{\eta(\mathbf{x})}_{\text{linear}}) = g^{-1}(\beta_0 + \beta_1 x^{(1)} + \dots + \beta_p x^{(p)})$$

## Components

- ▶ Linear function/predictor  $\eta$
- ▶ Link function  $g$  to map  $\mathbb{E}(Y | \mathbf{x})$  to linear scale
- ▶ Distribution of  $Y$  conditional on covariates  $\rightarrow$  loss function (unit deviance)

$g = \text{Id} \rightarrow \text{lin. Model}$

# Typical GLMs

Regression	Distribution	Range of $Y$	Natural link	Unit deviance
Linear	Normal	$(-\infty, \infty)$	Identity	$(y - \hat{y})^2$
Logistic	Binary	$\{0, 1\}$	logit	$-2(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$
Poisson	Poisson	$[0, \infty)$	log	$\sum 2(y_i \log(y_i / \hat{y}_i) - (y_i - \hat{y}_i))$
Gamma	Gamma	$(0, \infty)$	$1/x$ (typical: log)	$2((y - \hat{y})/\hat{y} - \log(y/\hat{y}))$
Multinomial	Multinomial	$\{C_1, \dots, C_m\}$	mlogit	$-2 \sum_{j=1}^m 1(y = C_j) \log(\hat{y}_j)$

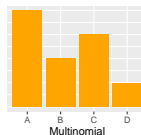
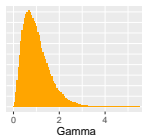
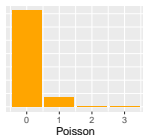
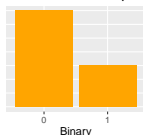
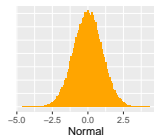
e.g. claim counts

which unit?  
(unit of  $Y$ )<sup>2</sup>

$$D \propto \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- Predictions?
- Log-Link? *log allows to talk about %*
- For binary  $Y$ : *all about percentage differences*  
 $\mathbb{E}(Y) = P(Y = 1) = p$

Empirical distributions of 100'000 values



right skewed  
light tails

- MSE  $\rightarrow$  Deviance
- Losses in ML?  
*look at Deviance of GLM*

Dev-0 - Deviance\_model

*Dev-0  
same with  $R^2$*

# Why GLM, not Linear Regression?

## Linearity assumption not always realistic

### 1. Binary $Y$ :

Jump from 0.5 to 0.6 success probability less impressive than from 0.89 to 0.99

### 2. Count $Y$ : Jump from $\mathbb{E}(Y)$ of 2 to 3 less impressive than from 0.1 to 1.1.

### 3. Right-skewed $Y$ :

Jump from 1 Mio to 1.1 Mio deemed larger than from 2 Mio to 2.1 Mio.

*factor of ten*

Logarithmic  $Y$  not possible in the first two cases

GLM solves problem by suitable link  $g$

$$\log(\mathbb{E}(Y|x))$$

*combs: often values of zero  $\rightarrow$  can't take log*

Further advantages?

*$\log(\text{of pred } 0) = \text{defined!}$*

*Link function squeeze*

*the output space  $\rightarrow [0, 1]$   
 $\rightarrow [0, \infty]$  etc*

*not normally exactly 0, but 0.001*

# Interpretation of Effects guided by Link

## Identity link

Like linear regression

## Log link

Like linear regression with log response

- ▶ Multiplicative model for response
- ▶ Now in mathematically sound way

## Logit link

- ▶ Additive model for  $\text{logit}(p)$
- ▶  $\text{logit}(p) = \log(\text{odds}(p)) = \log\left(\frac{p}{1-p}\right)$
- ▶ Remember:  $p = P(Y = 1) = \mathbb{E}(Y)$
- ▶ Multiplicative model for  $\text{odds}(p)$
- ▶ Coefficients  $e^{\beta} - 1 \approx 100\%\beta$  interpreted as odds ratios

## Examples with Insurance Claim Data

1. Poisson regression for claim counts
2. Binary logistic regression for claim (yes/no)

# Modeling Large Data

## As per 2023

- ▶ On normal laptops, we can model datasets up to 8 GB in size (1 Mio iris data)
- ▶ Cloud computing allows 1000 times more
- ▶ We focus on in-memory situations  
→ data fits in RAM

## Example

## Aspect and example technology

1. Data storage → Apache Parquet
2. Data loading → Apache Arrow
3. Preprocessing → data.table
4. Modeling → H2O

run-length encoding

datatypes preserved

like CSV

data is stored in RAM

can do in arrow

java bundle  
- superfast  
- cluster computing

fastest in-memory technology