



Advanced Data Science

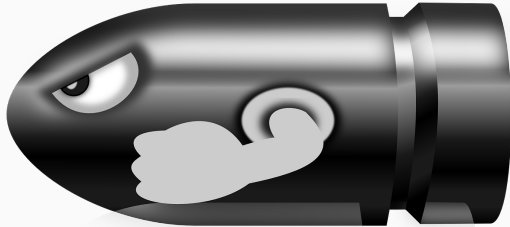
Lecture 6 : Generalised Linear Models

Carl Henrik Ek - che29@cam.ac.uk

22nd of November, 2021

<http://carlhenrik.com>

What is Machine Learning



What does Machine Learning do?



$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)p(\theta)}{p(\mathcal{D})}$$



- PAC Learning provides an useful foundation for abstracting learning
 - we need a lot more data than we think
 - there is no such thing as a universal learner
- We are doing a lot better in practice than we should

Access enormous inductive bias in what data to acquire

Access enormous inductive bias in what data to acquire

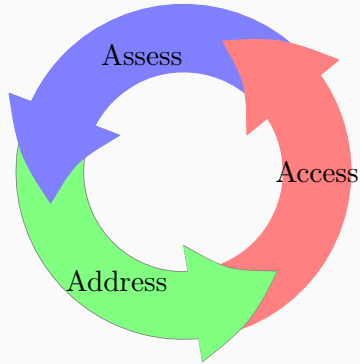
Assess human bias in what questions will probably be asked

Access enormous inductive bias in what data to acquire

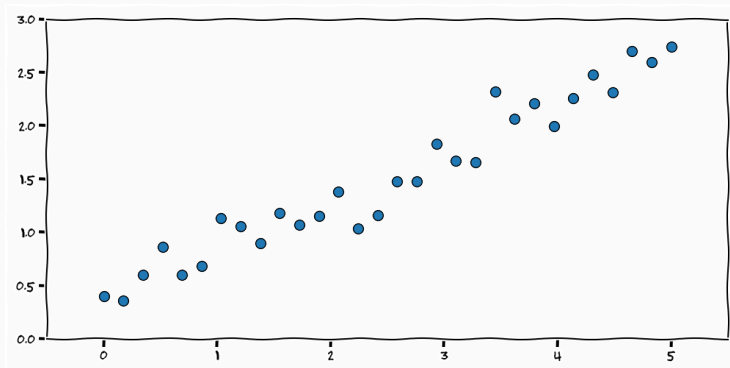
Assess human bias in what questions will probably be asked

Address "it is just curve fitting"

Requirements for Data Science



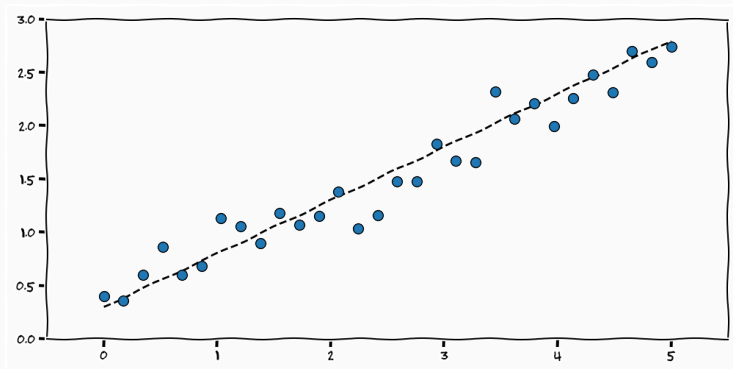
Generalised Linear Models



$\mathbf{x} \in \mathcal{X}$ explanatory variable

$y \in \mathcal{Y}$ response variable

Task *explain the response by the explanatory variables*



$$y_i = \sum_{j=1}^d \beta_j x_{ij} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2).$$

$$\mathbb{E}[y_i \mid \mathbf{x}_i] = \mathbb{E} \left[\sum_{j=1}^d \beta_j x_{ij} + \epsilon \right]$$

$$\begin{aligned}\mathbb{E}[y_i \mid \mathbf{x}_i] &= \mathbb{E} \left[\sum_{j=1}^d \beta_j x_{ij} + \epsilon \right] \\ &= \mathbb{E} \left[\sum_{j=1}^d \beta_j x_{ij} \right] + \mathbb{E}[\epsilon]\end{aligned}$$

$$\begin{aligned}\mathbb{E}[y_i \mid \mathbf{x}_i] &= \mathbb{E} \left[\sum_{j=1}^d \beta_j x_{ij} + \epsilon \right] \\ &= \mathbb{E} \left[\sum_{j=1}^d \beta_j x_{ij} \right] + \mathbb{E}[\epsilon] \\ &= \sum_{j=1}^d \beta_j x_{ij} + 0.\end{aligned}$$

$$y_i = \sum_{j=1}^d \beta_j x_{ij} + \epsilon,$$

$$y_i = \sum_{j=1}^d \beta_j x_{ij} + \epsilon,$$

$$y_i + \epsilon = \sum_{j=1}^d \beta_j x_{ij},$$

$$y_i = \sum_{j=1}^d \beta_j x_{ij} + \epsilon,$$

$$y_i + \epsilon = \sum_{j=1}^d \beta_j x_{ij},$$

$$\hat{y}_i = \sum_{j=1}^d \beta_j x_{ij},$$

$$y_i = \sum_{j=1}^d \beta_j x_{ij} + \epsilon,$$

$$y_i + \epsilon = \sum_{j=1}^d \beta_j x_{ij},$$

$$\hat{y}_i = \sum_{j=1}^d \beta_j x_{ij},$$

$$\hat{y}_i \sim \mathcal{N}(y_i, \sigma^2) = \mathcal{N} \left(\sum_{j=1}^d \beta_j x_{ij}, \sigma^2 \right),$$

$$g(\mathbb{E}[y_i \mid \mathbf{x}_i]) = \sum_{j=1}^d \beta_j x_{ij},$$

$g(\cdot)$ link function

$y \sim \mathcal{D}$ Exponential Dispersion Family

$\sum_{j=1}^d \beta_j x_{ij}$ Linear predictor

$$\mathbb{E}[y_i \mid \mathbf{x}_i] = g^{-1}\left(\sum_{j=1}^d \beta_j x_{ij}\right),$$

- The inverse of the *link* maps the linear predictor to the first moment of the response
- Linear regression the link is identity

¹<https://towardsdatascience.com/>

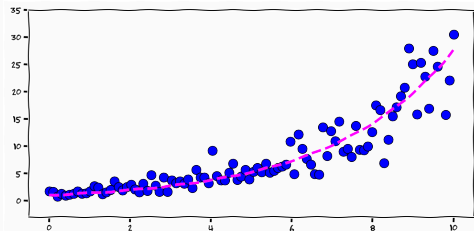
[glms-part-iii-deep-neural-networks-as-recursive-generalized-linear-URL](#)

$$\mathbb{E}[y_i \mid \mathbf{x}_i] = g^{-1}\left(\sum_{j=1}^d \beta_j x_{ij}\right),$$

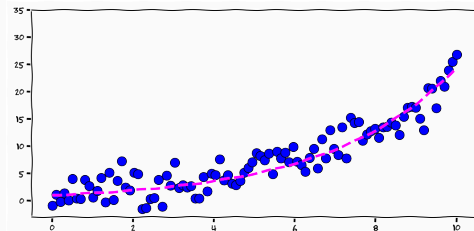
- The inverse of the *link* maps the linear predictor to the first moment of the response
- Linear regression the link is identity
- Looks an awful lot like a neural network¹

¹<https://towardsdatascience.com/>

Transformation vs GLM



$$\log(y_i) = \sum_{j=1}^d \beta_j x_{ij} + \epsilon_i$$
$$\epsilon_i \sim \mathcal{N}(0, \sigma^2)$$



$$\log(y_i) = \sum_{j=1}^d \beta_j x_{ij}$$
$$y_i \sim \mathcal{N}(0, \sigma^2)$$

$$f(y; \theta, \phi) = e^{\frac{\theta y - b(\theta)}{a(\phi)} + c(y, \phi)},$$

θ location parameter

ϕ scale parameter

i.i.d. we will assume that the data is drawn i.i.d.

²https://en.wikipedia.org/wiki/Exponential_dispersion_model

$$f(y; \mu, \sigma^2) = e^{\frac{\mu y - \frac{1}{2}\mu^2}{\sigma^2} - \frac{y^2}{2\sigma^2} - \frac{1}{2} \ln(2\pi\sigma^2)}$$

- $\theta = \mu$
- $\phi = \sigma^2$
- $b(\theta) = \frac{1}{2}\mu^2$
- $a(\phi) = \sigma^2$
- $c(y, \phi) = \frac{y^2}{2\sigma^2} - \frac{1}{2} \ln(2\pi\sigma^2)$

$$\mathbb{E}[y \mid \mathbf{x}] = \frac{\partial}{\partial \theta} b(\theta)$$
$$\mathbb{V}[y \mid \mathbf{x}] = a(\phi) \frac{\partial^2}{\partial \theta^2} b(\theta).$$

- Through a consistent parametrisation we can generalise the moment calculations

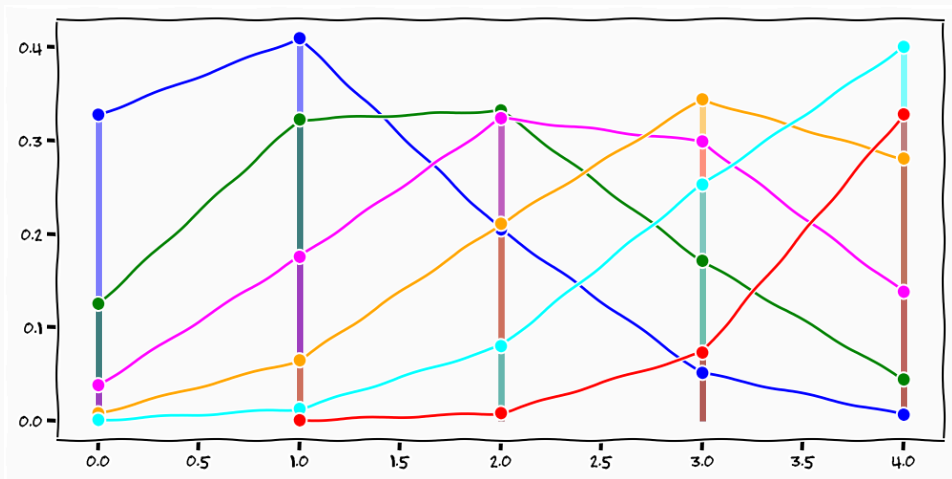
Code

```
import statsmodels.api as sm
m = sm.GLM(y, x, sm.families.Gaussian(sm.families.links.log()))
m_r = m.fit()
y_p = m_r.get_prediction(x_p).summary_frame(alpha=0.05)['mean']
```

Families Binomial, Gamma, Gaussian, InverseGaussian,
NegativeBinomial, Poisson, Tweedie

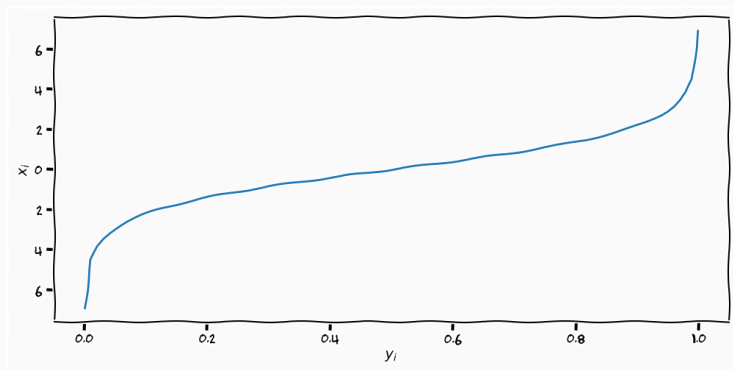
Link Functions CLogLog, LogLog, Log, Logit, NegativeBinomial,
Power, cauchy, identity, inverse_power,
inverse_squared, nbinom, probit

Binomial Distribution



$$g(y_i) = \beta_0 + \beta_1 x_{i1}$$
$$y_i \sim \text{Binom}(n, p)$$

- y_i is a frequency or odds
- need to pick a link function that limits to $y_i \in [0, 1]$



$$\text{logit}(y_i) = \log \left(\frac{y_i}{1 - y_i} \right)$$

$$\text{logit}(y_i) = \log \left(\frac{y_i}{1 - y_i} \right) = \beta_0 + \beta_1 x_{i1}$$

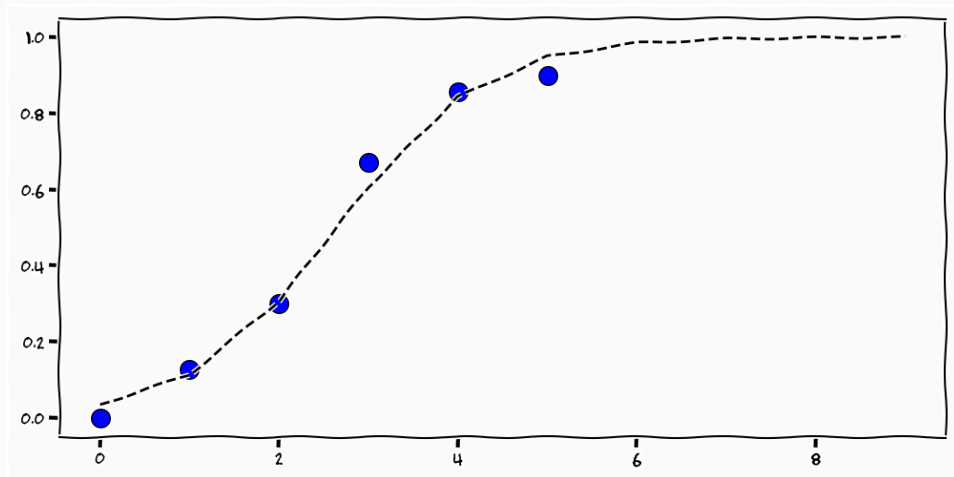
$$\frac{y_i}{1 - y_i} = e^{\beta_0 + \beta_1 x_{i1}}$$

$$y_i(1 + e^{\beta_0 + \beta_1 x_{i1}}) = e^{\beta_0 + \beta_1 x_{i1}}$$

$$\begin{aligned} y_i &= \frac{e^{\beta_0 + \beta_1 x_{i1}}}{1 + e^{\beta_0 + \beta_1 x_{i1}}} \\ &= \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{i1})}} \end{aligned}$$

Current	Trials	Response	Proportion
0	70	0	0.00
1	70	9	0.129
2	70	21	0.300
3	70	47	0.671
4	70	60	0.857
5	70	63	0.900

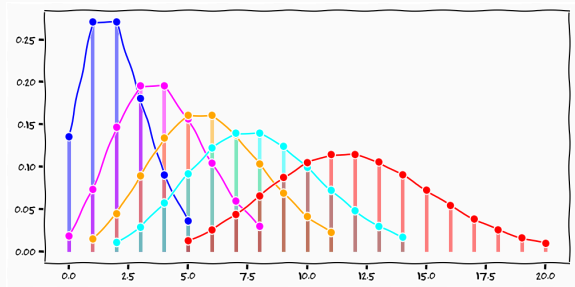
Logistic Regression



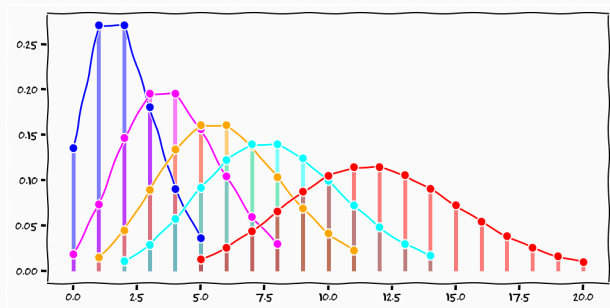
```
sm.GLM(y, X, sm.families.Binomial(sm.families.links.logit()))
```


Poisson Distribution

- Arrival times
- Website visitors
- Job cue for server
- Failures of product



Poisson Distribution



$$\text{Poisson}(y_i) = \frac{e^{-\lambda} \lambda^y}{y!}$$

$$\mathbb{E}[y_i] = \lambda$$

- Counts are positive so we need a positive link function

$$\log(\lambda_i) = \sum_{j=1}^d \beta_j x_{ij}$$

- Counts are positive so we need a positive link function

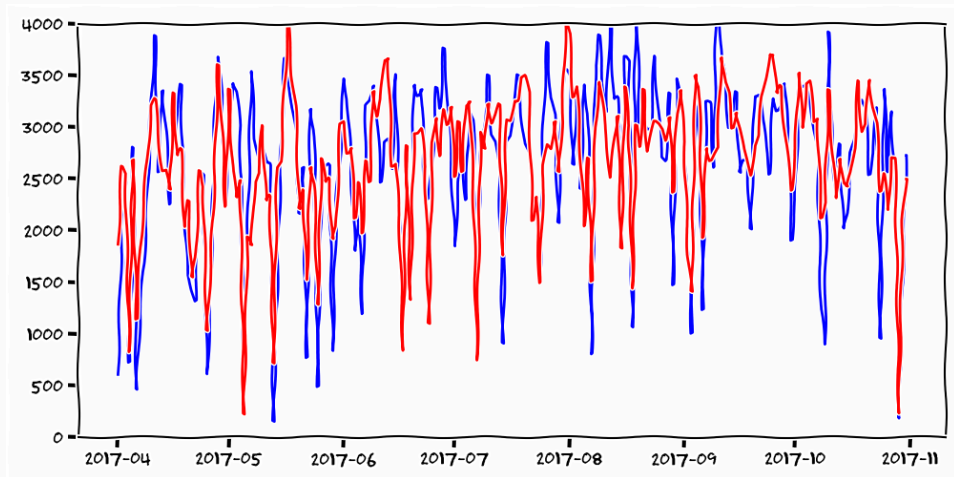
$$\log(\lambda_i) = \sum_{j=1}^d \beta_j x_{ij}$$

- Leads to the following model

$$p(y_i | x_i) = \frac{e^{\lambda_i} \lambda_i^{y_i}}{y_i!} = \frac{e^{\left(e^{\sum_{j=1}^d \beta_j x_{ij}}\right)} \left(e^{\sum_{j=1}^d \beta_j x_{ij}}\right)^{y_i}}{y_i!}$$

Day	Day of Week	Month	High Temp	Low Temp	Percipitation	Cyclists
1.0	5.0	4.0	46.0	37.0	0.00	606.0
2.0	6.0	4.0	62.1	41.0	0.00	2021.0
3.0	0.0	4.0	63.0	50.0	0.03	2470.0
4.0	1.0	4.0	51.1	46.0	1.18	723.0
6.0	3.0	4.0	48.9	41.0	0.73	461.0
...
1	31	10	54	44	0.00	2727

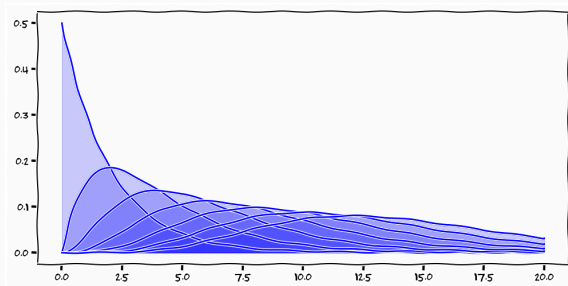
Poisson Regression



```
sm.GLM(y, X, family=sm.families.Poisson()).fit()
```

Gamma Distribution

- Waiting times for Poisson events
- Variance and mean connected

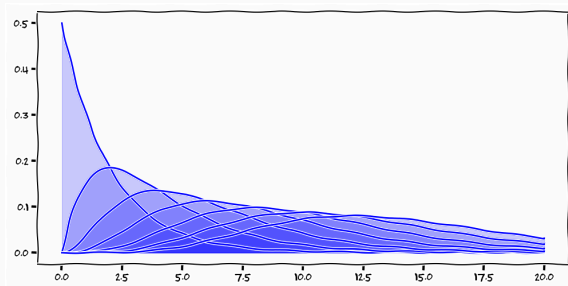


Gamma Distribution

$$\text{Gamma}(y_i) = \frac{1}{\Gamma(\phi)\theta\phi} y_i^{\phi-1} e^{-\frac{y_i}{\theta}}$$

$$\mathbb{E}[y_i] = \phi\theta$$

$$\mathbb{V}[y_i] = \phi\theta^2$$



- Exponential Dispersion Gamma

$$\text{Gamma}(y_i) = e^{\frac{y_i \theta_i - \log(-\frac{1}{\theta_i})}{\phi} + \frac{1-\phi}{\phi} \log(y_i) - \log(\Gamma(\phi^{-1}))}$$

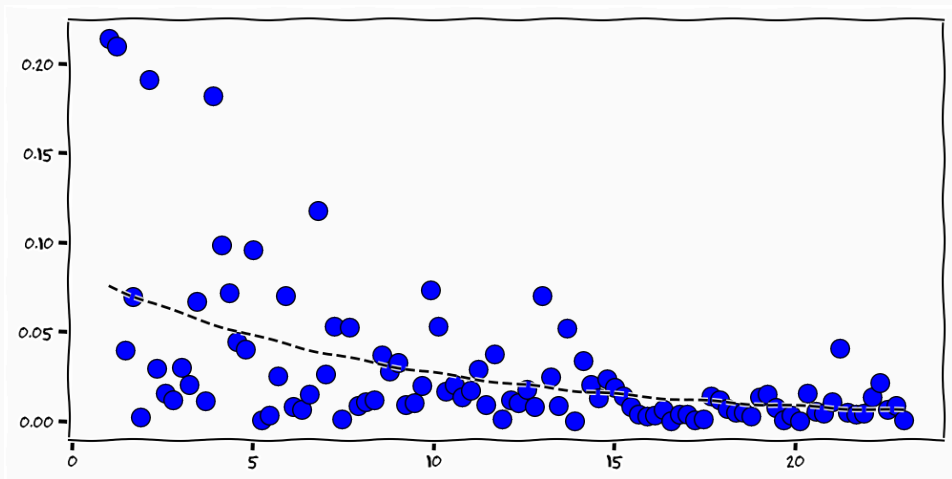
- If you "derive" the canonical link function from the distribution it should be,

$$-\frac{1}{\theta} = \sum_{j=1}^d \beta_j x_{ij}$$

- Gamma regression is most commonly used with \log as the link

$$\log(\mathbb{E}[y_i \mid \mathbf{x}_i]) = \sum_{j=1}^d \beta_j x_{ij}$$

Gamma Regression



Model	Response Variable	Link	Explanatory Variable
Linear Regression	Normal	Identity	Continuous
Logistic Regression	Binomial	Logit	Mixed
Poisson Regression	Poisson	Log	Mixed
ANOVA	Normal	Identity	Categorical
ANCOVA	Normal	Identity	Mixed
Loglinear	Poisson	Log	Categorical
Multinomial response	Multinomial	Generalized Logit	Mixed

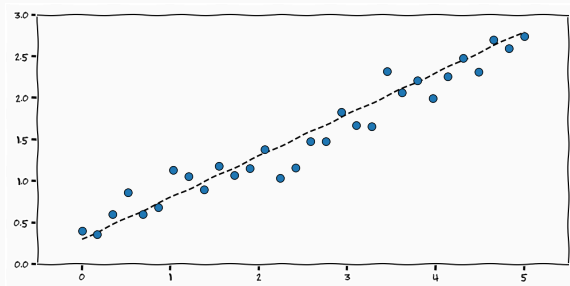
$$\hat{\beta} = \operatorname{argmax}_{\beta} \prod_{i=1}^N p(y_i \mid \beta, \mathbf{x}_i)$$

- In general gradient descent on log-likelihood
- For specific models there are tailored inference schemes

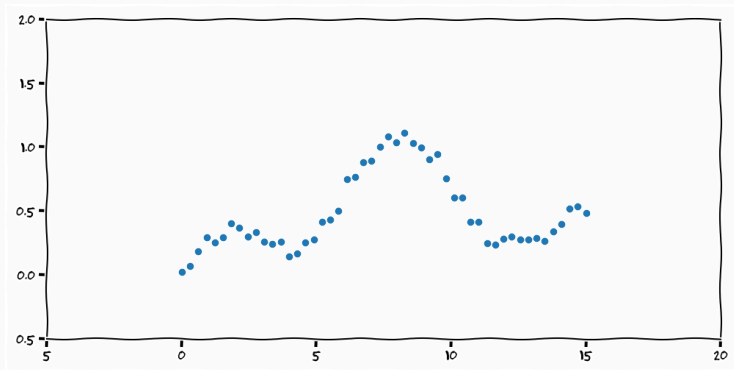
Design Matrix

Design Matrix

$$\mathbf{X} = \begin{bmatrix} x_0 & 1 \\ x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix}$$

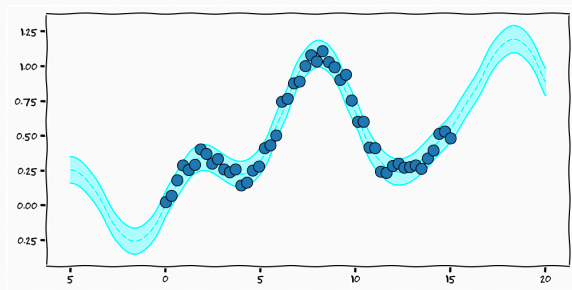


Non-Linear Function



$$y = 0.2 \sin(x) + \sin\left(\frac{x^2}{40}\right) + 0.05x$$

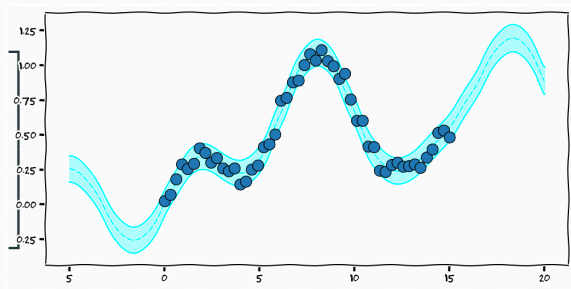
$$\mathbf{X} = \begin{bmatrix} \sin(x_0) & \sin(\frac{x_0^2}{40}) & x_0 \\ \sin(x_1) & \sin(\frac{x_1^2}{40}) & x_1 \\ \vdots & \vdots & \vdots \\ \sin(x_N) & \sin(\frac{x_N^2}{40}) & x_N \end{bmatrix}$$



$$\boldsymbol{\beta} = [0.2155, 0.4956, 0.0482]$$

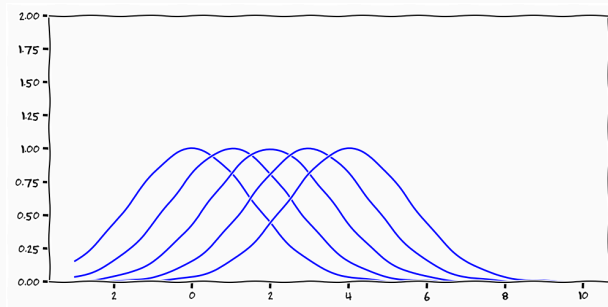
Over-parametrised matrix

$$\begin{bmatrix} \sin(x_0) & \sin(\frac{x_0^2}{40}) & x_0 & -\sin(x_0) \\ \sin(x_1) & \sin(\frac{x_1^2}{40}) & x_1 & -\sin(x_1) \\ \vdots & \vdots & \vdots & \vdots \\ \sin(x_N) & \sin(\frac{x_N^2}{40}) & x_N & \sin(x_N) \end{bmatrix}$$



$$\beta = [0.1078, 0.4956, 0.0482, 0.1078]$$

Localised Basis Function



$$g(\mathbb{E}[y_i \mathbf{x}_i]) = \sum_{j=1}^N \beta_j \phi(\mathbf{x}_j, \mathbf{x}_i), \quad \phi(\mathbf{x}_j, \mathbf{x}_i) = e^{-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)}{\ell^2}}$$

Regularisation

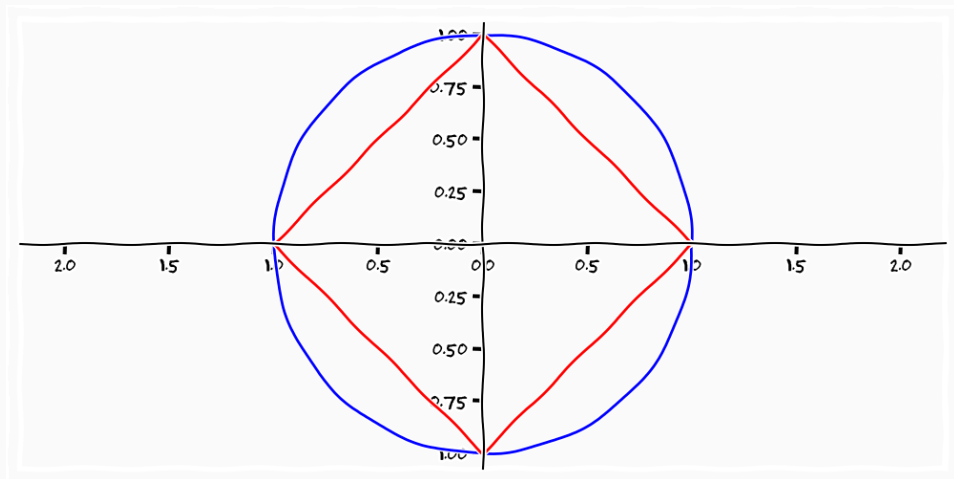
$$\hat{\beta} = \operatorname{argmax}_{\beta} \prod_{i=1}^N p(y_i \mid \beta, \mathbf{x}_i)$$

- Maximum Likelihood encodes no **preference** towards any solution
- Due to optimisation procedure we might get very different results

$$\hat{\boldsymbol{\beta}} = \operatorname{argmax}_{\boldsymbol{\beta}} \prod_{i=1}^N p(y_i \mid \boldsymbol{\beta}, \mathbf{x}_i) + \lambda \left(\sum_{j=1}^d \beta_j^p \right)^{\frac{1}{p}}$$

- Introduce inductive bias towards specific solutions
- Normally done using a norm

Ridge vs Lasso



Code

```
m.fit_regularized(alpha=0.10,L1_wt=0.0)
```

- L1_wt 0 \rightarrow Ridge, 1 \rightarrow Lasso
- alpha the penalty

Summary

Response Variable Distribution How is your response variable distributed?

Response Variable Distribution How is your response variable distributed?

Link Function How is the **scale** parameter of the distribution related to the explanatory variables

Response Variable Distribution How is your response variable distributed?

Link Function How is the **scale** parameter of the distribution related to the explanatory variables

Design Matrix What is the features of the explanatory variables?

Response Variable Distribution How is your response variable distributed?

Link Function How is the **scale** parameter of the distribution related to the explanatory variables

Design Matrix What is the features of the explanatory variables?

Regulariser What is the "preferred" solution?

- Can you split up the data by some criterion?




- Can you split up the data by some criterion?
 - localised GLM

- Can you split up the data by some criterion?
 - localised GLM
- Can you remove the effect of one model from data and then retrain on residual?

- Can you split up the data by some criterion?
 - localised GLM
- Can you remove the effect of one model from data and then retrain on residual?
- You will **not** be able to find the "perfect" model, but show that you can reason about these models!!

eof

References

-  Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.
-  McCullagh, P. and J. A. Nelder (1989). *Generalized Linear Models*. London, UK: Chapman Hall / CRC: Chapman Hall / CRC.
-  Weisberg, Sanford (2005). *Applied Linear Regression*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., nil.