Bachelor Wiskunde

*Bachelorscriptie*

# Priorkeuze in Bayesiaanse Polynomiale Regressiemodellen

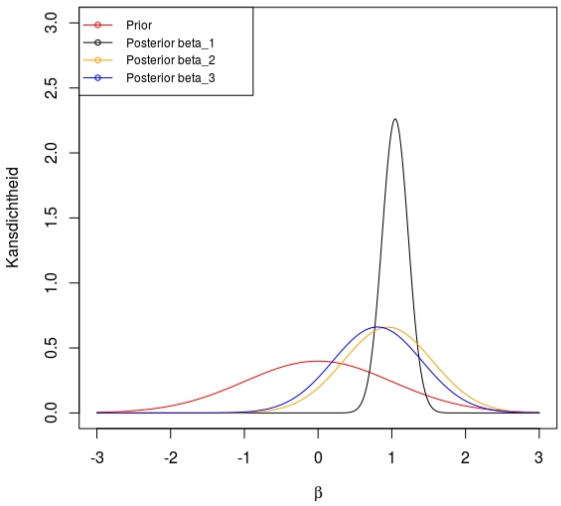
door

Thom Oosterhuis

30 juni 2018

Begeleider: T. Klausch

Tweede beoordelaar: M. van de Wiel




Afdeling Wiskunde
Faculteit der Bètawetenschappen

VU VRIJE UNIVERSITEIT AMSTERDAM

# Samenvatting

Een introductie in de Bayesiaanse regressie-analyse wordt gegeven. Deze wordt vergeleken met een aantal frequentistische regressiemethoden en schatters. Vervolgens wordt verder ingegaan op de prior keuze voor parameters in Bayesiaanse polynomiale regressiemodellen. Door middel van data simulaties worden priors en de hieruit voortkomende puntschatters op basis van de posterior met elkaar vergeleken aan de hand van de gekruisvalideerde MSE en de test MSE. Data van een bepaalde orde wordt gemodelleerd in een model van dezelfde orde en in een model van een hogere orde. De verschillende priors geven in het eerste model vergelijkbare schattingen. In het tweede model is er echter sprake van overfitting en zien we dat de normale prior met een optimale variantie deze overfitting het beste tegengaat en tegelijkertijd de andere coëfficiënten niet te laag schat. De andere priors geven hier schattingen die meer afwijken van de ware coëfficiënten.

Titel: Priorkeuze in Bayesiaanse Polynomiale Regressiemodellen
Auteur: Thom Oosterhuis, thom@vu.nl, 2563038
Begeleider: T. Klausch
Tweede beoordelaar: M. van de Wiel
Einddatum: 30 juni 2018

Afdeling Wiskunde
Vrije Universiteit Amsterdam
de Boelelaan 1081, 1081 HV Amsterdam
`http://www.math.vu.nl/`

# Inhoudsopgave

# 1 Inleiding

Gegeven een dataset bestaande uit een $n$-aantal datapunten $(X, Y) = (x_1, y_1), \ldots, (x_n, y_n)$, waarbij $X$ zowel één als meer dimensionaal kan zijn, willen we vaak een regressielijn vinden in de vorm van $Y = f(X)$, om zo iets te kunnen zeggen over hoe $Y$ van $X$ afhangt. Er zijn natuurlijk veel manieren om dit te doen. De bekendste aanpak is waarschijnlijk de simpele lineaire regressie, waarbij $X$ één dimensionaal is en we een functie van de vorm $Y = \beta_0 + \beta_1 X$ proberen te vinden. Het hellingsgetal $\beta_1$ en het punt waar de lijn de Y-as snijdt $\beta_0$, kunnen worden geschat met de methode van de kleinste kwadraten. Deze methode kent veel variaties, waarbij bijvoorbeeld de lineaire formule $f(X)$ kan worden vervangen door een functie $f(X_1, \ldots, X_k)$ die van meerdere variabelen afhangt of door een functie die ook polynomiale termen bevat.

Het bovenstaande probleem een uitspraak te doen over de afhankelijkheid van $Y$ van $X$ kan echter ook op een andere manier benaderd worden. De Bayesiaanse methode neemt weliswaar aan dat er een oplossing van de vorm $Y = f(X)$ bestaat, maar in plaats van een schatting te doen voor de coëfficiënten $\beta_0, \ldots, \beta_n$, neemt deze methode aan dat deze coëfficiënten een eigen kansverdeling hebben. Het probleem van het vinden van de juiste schattingen van de coëfficiënten, verandert dan in het vraagstuk van het vinden van de juiste kansverdelingen van de coëfficiënten. Deze andere visie op het statistische model vereist bovendien dat er een initiële kansverdeling voor de coëfficiënten moet worden gekozen, een zogeheten prior. Deze initiële keuze is, samen met de data, bepalend voor de uitkomst van de uiteindelijke kansverdeling. De keuze voor een prior is niet eenduidig en daarmee ontstaat er dus een subjectivistisch element in het statistisch model.

We zullen zien dat bepaalde prior keuzes deze subjectiviteit zullen beperken, terwijl anderen juist een grotere vrijheid tot interpretatie vooraf aan de statisticus geven. Dit project zal gaan over het vraagstuk van het kiezen van een prior en zijn parameterwaarden. We zullen hierbij naar polynomiale modellen kijken met één enkele onafhankelijke variabele. De data en de regressiemodellen zullen dus van de vorm $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_1^p = \mathbf{X}\vec{\beta}$ zijn. We zullen hiertoe allereerst beginnen met een wiskundige onderbouwing van het Bayesiaanse model en dit waar nodig geacht, vergelijken met het frequentistische model. Enkele voorbeelden zullen hierbij gegeven worden. Vanuit de theorie zal de onderzoeksvraag duidelijk gespecificeerd worden en zal de opzet naar het experiment waarin verschillende priors met elkaar worden vergeleken, beschreven worden. We zullen in verschillende simulaties priors met elkaar vergelijken. De resultaten van deze verschillende simulaties zullen tot slot besproken worden, om te kijken in hoeverre we de onderzoeksvraag kunnen beantwoorden.

# 2 Frequentistische regressie

## 2.1 Basis lineaire en polynomiale regressie

We willen iets over de relatie tussen $\mathbf{X} = (X_1, \ldots, X_k, \ldots, X_q)$ en $Y$ zeggen op basis van een $n$ aantal datapunten $(x_i, y_i)$. We nemen nu aan dat we meer data punten dan variabelen $X_k$ hebben. Dus

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k + \cdots + \beta_q X_q + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{2.1}$$

waarbij

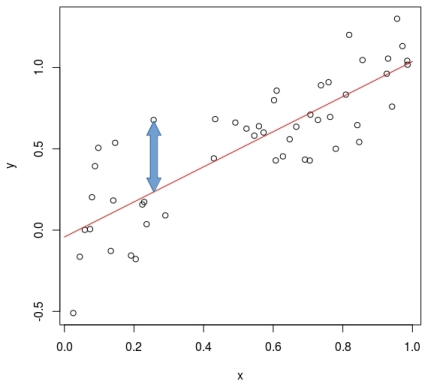
$$X_k = \begin{bmatrix} x_{(k,1)} & \ldots & x_{(k,i)} & \ldots & x_{(k,n)} \end{bmatrix}^T$$

met $n > q$. We spreken in dit geval van een overgedetimineerd probleem. Merk op dat $Y$ niet precies voorspeld kan worden door $X$. $Y$ bevat ook een een foutterm $\epsilon$. We nemen aan dat deze foutterm normaal verdeeld is rond nul met een bekende variantie $\sigma^2$. Voor $Y$ willen we dan een uitspraak doen over de verwachting van $Y$ gegeven de onafhankelijke variabelen $X_1, \ldots, X_q$ en de bijbehorende schattingen van de coëfficiënten $\beta_0, \ldots, \beta_q$. De verwachting voor een datapunt $y_i$ kunnen we dan noteren als $\hat{y}_i = \mathbb{E}(y_i | x_{i,1}, \ldots, x_{i,q}, \vec{\beta})$. Door de foutterm $\epsilon$ is het bijna onmogelijk en zeker ook onwenselijk een vergelijking te vinden die voor elke $X$ de precieze waarde voor $Y$ geeft. In plaats daarvan kiezen we ervoor een vergelijking te vinden die de voorspellingsfout, dat wil zeggen een nog te definiëren afstand tussen de verwachting $\hat{Y}$ en de daadewerkelijke observatie $Y_{\text{obs}}$, op een bepaalde manier minimaliseert. Een veel gebruikte maatstaf voor de totale voorspellingsfout is de som van het kwadraat van de residuen. Het residu definiëren we daarbij als de afstand tussen $\hat{Y}$ en $Y_{\text{obs}}$ over de $y$-as gemeten, zoals geschetst in figuur 2.1. We kiezen $f(X)$ dan zodanig dat de term $\sum_{i=1}^{n} (f(X_i) - Y_{\text{obs},i})^2$ wordt geminimaliseerd. Hierbij is $Y_{\text{obs},i}$ de geobserveerde waarde en $\hat{Y}_i = f(X_i)$ de voorspelde waarde.

Als we aannemen dat de regressievergelijking van de vorm $f(Y) = \mathbf{X}\vec{\beta}$ is, hebben we een vast vorm oplossing voor de schatting van $\beta$,

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T y. \tag{2.2}$$

Dit lineaire model met een $q$-aantal variabelen laat zich makkelijk uitbreiden naar een polynomiaal model waarbij elke $X_k$ voorkomt in hogere orde $X_k + X_k^2 + \cdots + X_k^p$. Dit




Figuur 2.1: Datapunten $(X, Y)$, $X = X_1$. De blauwe pijl geeft het residu tussen een datapunt en de regressielijn aan.

model is dan van de vorm

$$\begin{aligned}
\hat{Y} = \beta_0 + \beta_{1,1}X_1 &+ \beta_{1,2}X_1^2 + \cdots + \beta_{1,j}X_1^j + \cdots + \beta_{1,p}X_1^p + \\
\beta_{2,1}X_2 &+ \beta_{2,2}X_2^2 + \cdots + \beta_{2,j}X_2^j + \cdots + \beta_{2,p}X_2^p + \\
\cdots + & \\
\beta_{k,1}X_k &+ \beta_{k,2}X_k^2 + \cdots + \beta_{k,j}X_k^j + \cdots + \beta_{k,p}X_k^p + \\
\cdots + & \\
\beta_{q,1}X_q &+ \beta_{q,2}X_q^2 + \cdots + \beta_{q,j}X_q^j + \cdots + \beta_{q,p}X_q^p \\
= \mathbf{X}\vec{\beta}. &
\end{aligned}$$

We hebben nu een model waarbij de afhankelijke variabele door een $q$-aantal onafhankelijke variabele wordt voorspeld. Elk van deze variabele komt vervolgens in de vergelijking voor van orde 1 tot en met $p$. Later zullen we bij de verschillende datasimulaties en het kiezen van priors voor deze data ons alleen richten op één enkele variabele $X_1$ die in de regressievergelijking verschillende orden heeft: $Y = \beta_0 + X_1^1 + \cdots + X_1^p$. Dit vooral om de simulaties overzichtelijk en begrijpbaar te houden, de theorie en het achterliggende

principe van de regressievergelijking en het schatten van haar coëfficiënten verandert echter niet wanneer men het model naar meerdere variabelen uitbreidt.

## 2.2 Ridge schatter

Ridge regressie lijkt op de standaard lineaire regressie waarbij de coëfficiënten worden berekend met vergelijking (2.2). De toevoeging is dat er een *penalty* of *bestraffing* op de waarden van de regressiecoëfficiënten wordt gezet, zodat deze kleiner blijven of eerder naar een bepaalde waarde gaan. In plaats van het minimaliseren van de Residual Sum of Squares wordt de volgende vergelijking geminimaliseerd over $\beta$

$$\beta_{ridge} = \text{argmin}_\beta\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x + ij\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2\} \tag{2.3}$$

(Van Wieringen, 2018). In matrix vorm

$$\text{RSS}(\lambda) = (y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta. \tag{2.4}$$

De oplossing voor de ridge regressiecoëfficiënten is nu gelijk aan

$$\hat{\beta}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T y. \tag{2.5}$$

Dit lijkt op de oplossing voor het standaard lineaire model (2.2) met het verschil dat er een term $\lambda\mathbf{I}$ wordt opgeteld bij $\mathbf{X}^T\mathbf{X}$. Dit zorgt er ook voor dat wanneer de matrix $\mathbf{X}^T\mathbf{X}$ niet inverteerbaar is, de matrix $\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}$ dit wel is, zelfs als $\mathbf{X}^T\mathbf{X}$ geen volledige rank heeft. Een andere motivatie dan het bestraffen van bepaalde (hoge) parameterwaarden om de Ridge schatter te gebruiken kan daarom ook zijn dat men een oplossing wil vinden voor een niet inverteerbare matrix. Dit is bijvoorbeeld het geval bij een ondergedetimineerd probleem, waarbij er meer onafhankelijke variabelen $q$ dan respondenten $n$ zijn, iets dat bijvoorbeeld bij onderzoek naar genexpressie voor kan komen. De Ridge regressie kan ook worden afgeleid als het gemiddelde van een posterior verdeling met een bepaalde prior (Hastie, 2001).

# 3 Bayesiaanse regressie

## 3.1 Filosofie

In het Bayesiaanse model wordt niet alleen de data $Y$ maar ook de parameter $\theta$ volgens welke de data verdeeld is, als een willekeurige variabele gezien. Dit betekent dat $\theta$ geen vaste waarde heeft, maar een eigen kansverdeling. Het parametriseren van de data gebeurt daarom door het bepalen van een kansverdeling van de parameter $\theta$. We kiezen hiervoor eerst een a-priori verdeling van de parameter: $\Pi : \mathscr{G} \to [0, 1]$. Op basis van deze prior en de data kunnen we dan de posterior verdeling kiezen, de conditionele verdeling $\Pi_{\theta|Y} : \mathscr{G} \times \mathscr{Y} \to [0, 1]$. Onder bepaalde voorwaarde uit de maattheorie kunnen we de posterior als volgt uitdrukken:

$$\Pi(\mathscr{C} \in B|Y) = \int_B p_\theta(Y) d\Pi(\theta) \Big/ \int_\Theta p_\theta(Y) d\Pi(\theta) \tag{3.1}$$

(Kleijn, 2018). Hierbij is $\mathscr{C}$ een gebeurtenis uit de verzameling van alle mogelijke gebeurtenissen $B$, $Y$ de data en $\Theta$ de parameterruimte, zodat $\theta \in \Theta$.

Met behulp van de data en een priorverdeling voor deze parameters bepalen we de posterior verdeling van de parameters. Het model voor de regressie-analyse ziet er dan als volgt uit:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_{1,1} & \dots & x_{1,q} \\ \vdots & & \\ x_{n,1} & \dots & x_{n,q} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_q \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}, \ \epsilon_i \sim \mathcal{N}(0, \sigma^2), \tag{3.2}$$

$$\beta_j \sim \pi(\theta).$$

We nemen vaak aan dat $\sigma^2$ bekend is. Het doel van de Bayesiaanse regressie is nu het vinden van de juiste verdeling voor $\beta_j$.

**Voorbeeld** (Kleijn, 2018) Zij $X_1, \dots, X_n$ een identiek onafhankelijk verdeeld sample van een normale verdeling $\mathcal{N}(0, \sigma^2)$, $\sigma^2$ onbekend. We willen een posterior dichtheid voor $\sigma^2$ berekenen ten opzichte van de Lesbesgue maat op $[0, \infty)$. We nemen een prior voor de inverse van de variantie en substitueren naar een nieuwe variabele om de berekening makkelijker te maken: $\theta := \frac{1}{\sigma^2} \sim \Gamma(\alpha, \beta)$. We beschouwen alleen de data en de parameter

afhankelijkheid en laten constante termen weg, met behulp van (3.1) vinden we

$$\begin{aligned}
\Pi(\theta|X_1,\ldots,X_n) &\propto \prod_{i=1}^{n} p_\theta(x_i)\pi(\theta) \\
&\propto \prod_{i=1}^{n} \frac{\theta}{\sqrt{2\pi}} \exp\big(\frac{\theta x_i^2}{2}\big) \frac{\beta^\alpha}{\Gamma(\alpha)} \theta^{\alpha-1} \exp\big(-\beta\theta\big) \\
&\propto \theta^{\frac{n}{2}\alpha-1} \exp\big(\frac{\theta}{2}\sum_{i=1}^{n} x_i^2\big) \\
&\sim \Gamma(\tilde{\alpha},\tilde{\beta}), \tilde{\alpha} = \frac{n}{2}+\alpha, \tilde{\beta} = \beta + \frac{1}{2}\sum_{i=1} n x_i^2
\end{aligned}$$

## 3.2 De prior en de posterior

### 3.2.1 Puntschatters

De posterior verdeling van de parameter $\theta$ heeft een duidelijke interpretatie. In de praktijk kan het soms ook nuttig zijn op basis van deze verdeling een puntschatting $\hat{\theta}$ voor $\theta$ te maken. Verschillende puntschatters zijn de posterior mean, de posterior mediaan en de maxium a-posteriori (MAP) schatter (Kleijn, 2018). De MAP schatter is het punt $\theta$ waar de posterior kansdichtheid haar hoogste waarde aanneemt, het maximum van de posterior

$$\pi(\hat{\theta}_{MAP}|Y) = \sup_{\theta\in\Theta} \pi(\theta|Y).$$

Dit betekent dat de MAP-schatter de volgende afbeelding over $\theta$ maximaliseert:

$$\Theta \to \mathbb{R} : \theta \mapsto \prod_{i=1}^{n} p_\theta(X_i)\pi(\theta).$$

Indien de prior $\pi(\theta)$ uniform is, is dit equivalent aan het berekenen van de maximum likelihood schatter. Indien de prior niet uniform is, is het berekenen van de MAP-schatter gelijk aan het berekenen van een 'bestrafte' likelihood. De Ridge schatter is een voorbeeld van een dergelijke frequentistische schatter die een 'bestrafte' maximum likelihood maximaliseert over $\theta$. We hebben eerder al gezien dat de Ridge schatter door deze bestraffing schattingen naar nul krimpt. Wanneer we in de Bayesiaanse situatie een normale prior rond nul nemen en de MAP schatter nemen van de posterior die we hiermee krijgen, is deze schatting $\theta_{\text{MAP}}$ gelijk aan de Ridge schatter.

### 3.2.2 Paren van prior en posterior

Voor bepaalde combinaties van priors en likelihoods heeft de posterior een vaste vorm. Een goed voorbeeld hiervan is het lineaire model $Y = \beta_0 + \beta_1 X$ waarbij we $\beta = [\beta_0, \beta_1]^T$ willen bepalen en hiervoor een normale prior $\mathcal{N}(\mu_0, \sigma_0)$ kiezen. De posterior verdeling

is dan ook een normale verdeling met de volgende parameterwaarden (Bishop, p.153, 2006):

$$\begin{aligned} p(\beta \mid y) &= \mathcal{N}(\beta \mid \mu_N, \sigma_N), && (3.3)\\ \mu_N &= \sigma_N(\sigma_0^{-1}\mu_0 + \gamma \mathbf{X}^T y), && (3.4)\\ \sigma_N^{-1} &= \sigma_0^{-1} + \gamma \mathbf{X}^T \mathbf{X}. && (3.5) \end{aligned}$$

Hierbij is $\gamma$ de precisie factor waarvoor we in de voorbeelden en simulaties een vaste waarden zullen kiezen. Omdat de posterior Gausisch is, kunnen we makkelijk de MAP-schatter bepalen, deze is immers gelijk aan $\mu_N$.

Zowel bij priors met een normale verdeling als priors met een gamma verdeling zagen we hierboven al dat deze ook weer posteriors uit dezelfde familie van kansverdelingen geven. Een bepaalde verdeling van de data gecombineerd met een bepaalde priorkeuze levert in deze gevallen altijd een posterior van een bepaalde vorm op. Een dergelijke combinatie van prior en posterior noemen we een *conjugate family.*

## 3.3 De voorspellende posterior verdeling

In de praktijk willen we vaak voorspellingen kunnen maken voor een onbekende $Y$ op basis van de onafhankelijke variabelen $x_i$. We kunnen hiervoor de voorspellende kansverdeling gebruiken (Bishop, p.156, 2006):

$$p(y|y, \alpha, B) = \int p(y|\beta, B)p(\beta|y, \alpha, B)d\beta \tag{3.6}$$

Hierbij is $p(\beta|y)$ als in (3.3). Invullen van deze functie geeft de volgende vaste vorm

$$\begin{aligned} p(y|x, y, \alpha, B) &= \mathcal{N}(y|\mu_N^T x, \sigma_N^2(x)),\\ \sigma_N^2(x) &= \frac{1}{B} + \mathbf{X}^T S_N \mathbf{X} \end{aligned}$$

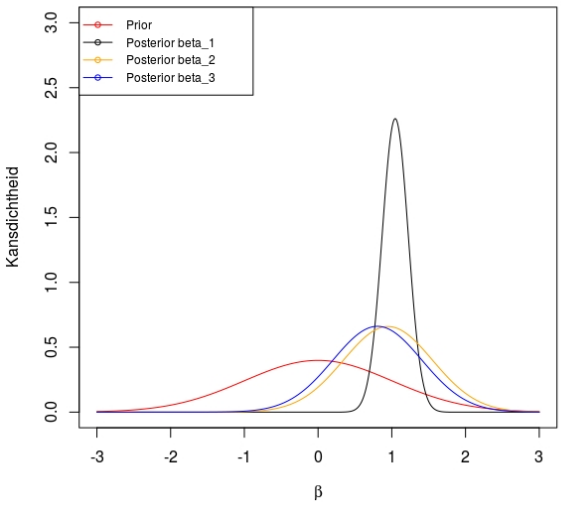
### 3.3.1 Voorbeeld

We genereren een sample van de vorm $Y = \beta_0 + \beta_1 X + \beta_2 X^2$ met $\beta_0 = 1, \beta_1 = 1, \beta_2 = 1.$ en een foutterm $\epsilon \sim \mathcal{N}(0, \sigma^2), \sigma^2 = 1,$ Op basis van de data $X, Y$ en drie prior verdelingen voor $\beta_0, \beta_1$ en $\beta_2$ gaan we nu de posterior verdelingen van $\beta_0$ en $\beta_1$ schatten.

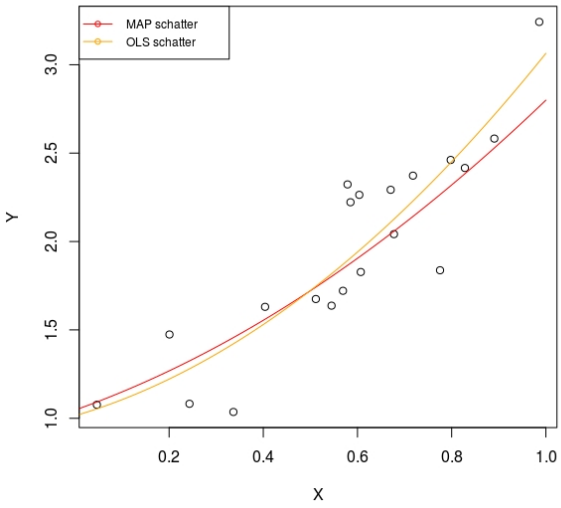
We kiezen voor alle drie de parameters een normale verdeling als prior met parameterwaarden $\mu_0 = 0, \sigma_0^2 = 1.$

$$\beta_0 \sim \mathcal{N}(0, 1), \beta_1 \sim \mathcal{N}(0, 1), \beta_2 \sim \mathcal{N}(0, 1)$$

We maken gebruik van vergelijking (3.1) en omdat de normale verdeling een conjugate family met zichzelf vormt, kunnen we gebruik maken van de oplossingen in vaste vorm zoals gegeven door vergelijking (3.4) en (3.5).




Figuur 3.1: Prior en posterior




Figuur 3.2: Data met regressielijnen voor MAP schatter en OLS schatter.

Invullen van de vergelijking, waarbij we $\gamma = 1$ voor de precisie term kiezen, geeft de volgende oplossingen:

$$\mu_{Nieuw} = \hat{\beta}_{MAP} = \begin{bmatrix} 1.05 \\ 0.95 \\ 0.80 \end{bmatrix},$$

$$\sigma_{Nieuw} = \begin{bmatrix} 0.18 & -0.21 & -0.038 \\ -0.21 & 0.60 & -0.34 \\ -0.038 & -0.34 & 0.60 \end{bmatrix}.$$

We merken op dat de MAP schatter gelijk is aan $\mu_{Nieuw}$ per definitie van de normale verdeling. Figuur 3.1 laat de prior en posterior verdelingen van de $\beta$ coëfficiënten zien. We zien dat de prior rond nul verdeeld is. Na het wegen van de data aan de hand van de prior krijgen we posteriors die rond hogere waarden zijn geconcentreerd en bovendien ook een kleinere variantie hebben. We kunnen hieruit opmaken dat de te schatten parameters met een grote waarschijnlijkheid een waarde aannemen boven nul. Voor het regressiemodel betekent dit dat $Y$ kan worden voorspeld aan de hand van $X$. De regressielijn op basis van de MAP schatters is weergeven in figuur 3.2, met daarbij ook de regressielijn van de OLS schatter. We zien dat, hoewel het verschil klein is, de regressielijnen duidelijk andere coëfficiënten hebben.

# 4 Het kiezen van priorverdelingen in polynomiale regressiemodellen

## 4.1 MSE, bias & variantie

De Mean Squared Error is een maatstaf voor de kwaliteit van een schatter. De MSE meet het gemiddelde van de kwadraten van de fouten, dat wil zeggen het verschil tussen de schatter en het geschatte. Algemeen geldt dat de MSE wordt gegeven door $E((\hat{\theta}-\theta)^2)$. We kunnen de MSE bovendien herschrijven

$$\begin{aligned} \mathrm{MSE}(\hat{\theta}) &= E((\hat{\theta}-\theta)^2) \\ &= E(\hat{\theta}-\theta) + E((\hat{\theta}-E(\hat{\theta}))^2) \\ &= \mathrm{Bias}(\hat{\theta}) + \mathrm{Var}(\hat{\theta}). \end{aligned}$$

Op basis van onze data en onze schattingen kunnen we een schatting maken voor de MSE:

$$\widehat{\mathrm{MSE}} = \frac{1}{k}\sum_{k=1}^{K}(Y_k - \hat{Y}_k)^2$$

Ook voor de bias en de variantie kunnen we op basis van onze data schattingen maken. Hierbij wordt de variantie gegeven door

$$\mathrm{Var}_\theta = \frac{1}{K}\sum_{k=1}^{K}(\theta_{N,k} - \bar{\theta}_N)^2$$

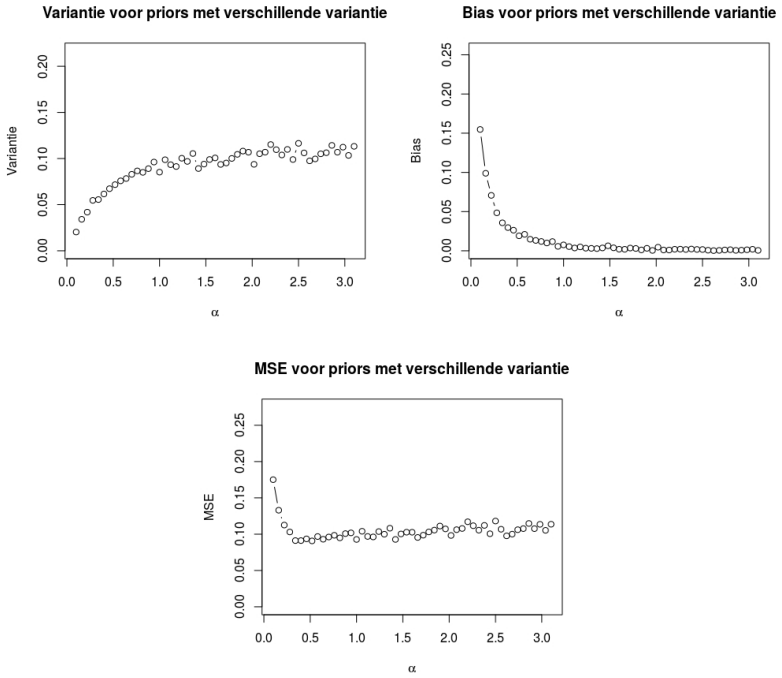
en de bias door

$$\mathrm{Bias}_\theta = |\bar{\theta}_{N,k} - \theta|.$$

Een goede schatter zal een lage MSE hebben en dus zowel een lage bias als een lage variantie. In de praktijk zien we echter vaak dat wanneer wij een schatter zo construeren dat deze een heel lage bias heeft, de variantie van deze schatter juist relatief groot zal zijn. Andersom zal een schatter met een lage variantie juist weer een grotere bias hebben. Voor het construeren van een goede schatter, zal dus een afweging tussen deze twee gemaakt moeten worden. Dit probleem staat in de literatuur ook wel bekend als de 'bias variance trade-off'.

Door middel van een experiment zullen we kijken hoe deze uitruil er uit ziet voor een Bayesiaanse schatting met een normale prior voor de regressiecoëfficiënt van een simpel lineair model $Y = \beta_0 + \beta_1 X$. We nemen dan telkens verschillende waarden voor de variantie $\alpha$ van de prior. Vervolgens simuleren we data de data $Y = 0 + \beta_1 X$ voor een zekere

$\beta_1 \in (0, 1)$. In dit voorbeeld kiezen we $\beta_1 = 0.8$. Voor de gegenereerde data berekenen we vervolgens de posterior distributie, op basis van een normale prior distributie met variantie $\sigma_0 = \alpha$ en gemiddelde $\mu_0 = 0$. We gebruiken hiervoor vergelijkingen (3.4) en (3.5). Voor deze trekking berekenen we de MSE, de bias en de variantie. Bovenstaande trekking herhalen we een $K$-aantal keer, waarbij we $K = 100$ kiezen. De uiteindelijke MSE, bias en variantie voor de betreffende waarden van $\alpha = \alpha_1$ worden nu gegeven door het gemiddelde over de $K$ iteraties. We kiezen nu $\alpha = \alpha_2$ en herhalen het hele bovenstaande proces. De resultaten voor de verschillende waarden van $\alpha$ staan weergeven in figuur 4.1.

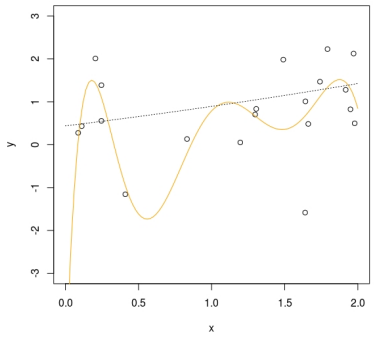



Figuur 4.1: Gemiddelde variantie, bias en MSE van schattingen voor 100 datasets voor verschillende waarden van $\alpha$.

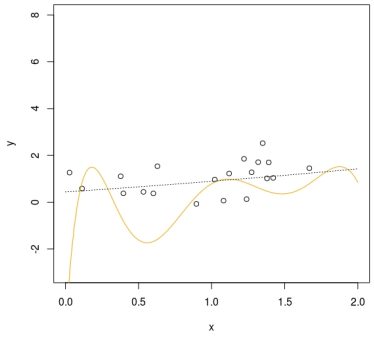
## 4.2 Overfitting

In de praktijk weten we vaak niet wat de ware orde is waarin de onafhankelijke variabelen de afhankelijke variabele voorspellen. Het kan aantrekkelijk lijken in een model een hoge orde $p$ te kiezen, omdat dit vaak tot een betere 'fit aan de data' en dus een lagere MSE leidt. Het volgende voorbeeld laat echter zien dat er ook een 'te goede' fit met de regressielijn en de data kan zijn, waardoor de regressie lijn teveel afhankelijk is van één specifiek sample. We genereren data van orde twee: $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon, \epsilon \sim \mathcal{N}(0, 1)$, waarbij we een samplegrootte van $n = 20$ nemen. Vervolgens bepalen we met behulp van Ordinair Least Squares zoals in vergelijking (2.2) een regressielijn van orde zeven. Figuur 4.2 laat zien dat de regressielijn goed bij de data past, bovendien is de MSE relatief laag,

namelijk 0.61. We genereren nu een drietal samples met hetzelfde polynomiale verband. We hopen te zien dat deze nieuwe samples ook goed aan de zojuist gevonden regressielijn passen. Immers, de sample data is afkomstig uit dezelfde 'ware' populatie. Figuur 4.3 laat zien dat dit echter niet het geval is en dat ook de MSE van deze drie samples met de eerder gevonden schatting voor de $\beta$, nu een stuk hoger ligt. Dit is te verklaren doordat de orde van de regressielijn te hoog was. Willekeur uit het eerste sample werd hierdoor onterecht niet als zodanig beschouwd, maar werd ook gemodelleerd met behulp van een polynoom van hoge orde. We zien dus dat een hoge orde regressielijn met een lage MSE voor een specifiek sample een slecht model kan geven.




Figuur 4.2: Eerste datasample met regressielijn van zevende orde, MSE = 0.61.




Figuur 4.3: Tweede nieuwe datasample met zelfde regressielijn van zevende orde geschat met het eerste sample, test MSE = 2.62.

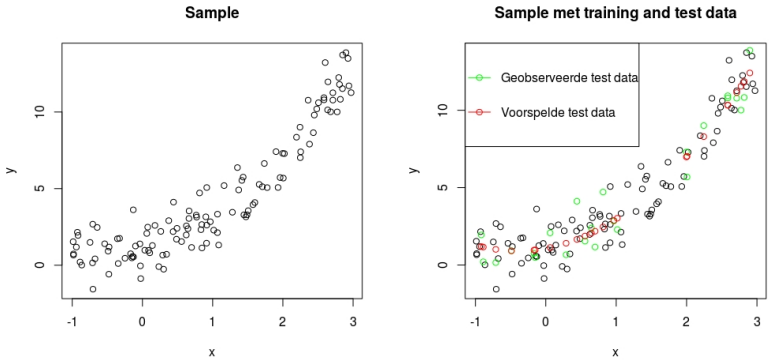
## 4.3 Kruisvalidatie

Kruisvalidatie is een manier om verschillende schattingen op basis van de data te maken en deze schattingen tegelijkertijd ook te testen. Hiertoe wordt de data opgedeeld in $K$ partities. Op basis van de $K-1$ partities - de trainingdata - wordt een schatting gemaakt

voor de te schatten parameters. Vervolgens wordt deze schatting met de andere partitie - de testdata - gecontroleerd. De schatting op basis van de testdata waarbij de $k^{\text{de}}$ partitie is weggelaten wordt weergeven met $\hat{f}^{-k}$. De schatting voor het datapunt $x_i$ uit de testdata wordt dan gegeven door $\hat{f}^{k(i)}(x_i)$. Met een verliesfunctie $L$, bijvoorbeeld de MSE, kan dan het de afstand tussen deze schatting en de observatie worden bekeken. De totale gekruisvalideerde MSE over alle partities wordt dan weergeven door vergelijking (4.2) (Hastie et al., p.212, 2001). Een voorbeeld van het opsplitsen van een dataset in training en testdata wordt geïllustreerd in figuur (4.4). We zien hier dat de groene datapunten tot de testdata behoren. Op basis van de schatting van de andere punten en de $X$ coördinaten van deze testdata wordt een voorspelling gemaakt voor de testdata, deze zijn als rode punten weergeven in de figuur.

$$\hat{\text{MSE}}_{\text{K.V.}}(\hat{f}^{-k}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i)) \tag{4.1}$$

$$= \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}^{-k(i)}(x_i))^2 \tag{4.2}$$




Figuur 4.4: Illustratie van kruisvalidatie.

## 4.4 Bestraffen en krimpen

Een statisticus zal bij het uitvoeren van een regressie-analyse vooraf niet alleen moeten bepalen welke onafhankelijke variabelen hij meeneemt in de analyse, maar ook de verschillende ordes van deze variabelen. Bij het uitvoeren van de regressie-analyse kan een overfit ook deels voorkomen worden. Een manier om dit te doen is het 'bestraffen' van bepaalde (hoge) schattingen. In plaats van de MSE te minimaliseren, wordt dan de som van de MSE en een straffunctie geminimaliseerd. Wanneer men deze straffunctie zo kiest dat deze hoger is voor hoge schattingen, worden de schattingen daardoor naar nul toe gekrompen.

Zowel binnen de frequentistische benadering als in de Bayesiaanse zijn er verschillende manieren om een dergelijke bestraffing van (hoge) schattingen toe te passen. Binnen de eerste benadering kan bijvoorbeeld een Least Absolute Shrinkage and Selection Operator(LASSO) regressie-analyse of de al eerder besproken Ridge regressie worden gebruikt. De Bayesiaanse benadering geeft over het algemeen meer mogelijkheden voor het bestraffen van bepaalde waarden van de schatting. Immers, per definitie wordt er gebruik gemaakt van een prior verdeling voor de te schatten variabele. Deze prior verdeling geeft dan een weging aan bepaalde observaties. Wanneer bijvoorbeeld lage waarden voor de schatting gewenst zijn, kan worden gekozen voor een prior die veel massa geeft aan lage waarden en weinig massa aan hoge waarden.

## 4.5 Mogelijkheden voor de priorverdeling

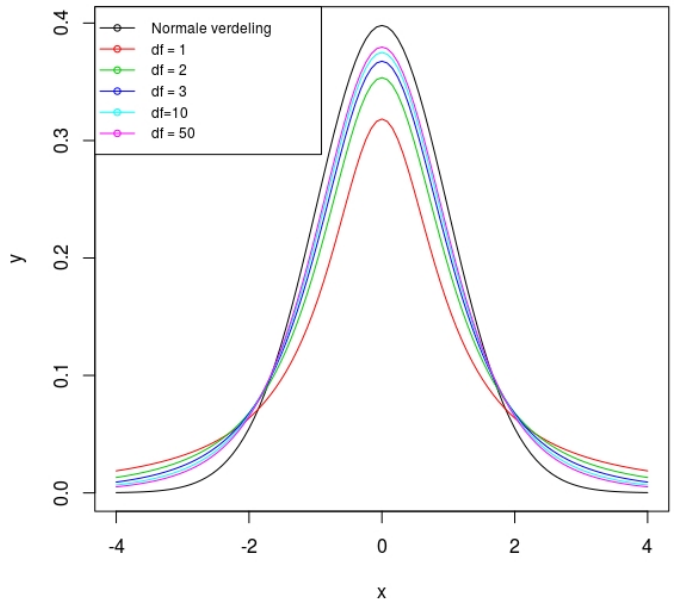
We zullen twee soorten priors bekijken voor de coëfficiënten $\beta_j$. Allereerst een prior met de normale verdeling met een gemiddelde nul en een variantie van $\alpha^{-1}$: $\beta_j \sim \mathcal{N}(0, \alpha^{-1})$. Deze prior is rond nul geconcentreerd en zal dus voor $\beta_j$ een posterior geven die deels naar nul is gekrompen. De mate waarin de posterior naar nul wordt gekrompen hangt natuurlijk sterk af van de variantie van de prior. Een normale prior met een heel lage variantie zal de posterior veel meer naar nul krimpen dan een normale prior met een hogere variantie. Deze laatste komt namelijk al in de buurt van een uniforme prior die even zwaar weegt aan elke observatie. We zullen daarom verschillende normale priors rond nul gebruiken en kijken of er een optimale variantie bestaat, die voor een groot aantal datasets steeds een goede afweging maakt tussen te hoge schattingen en krimpen.

Als tweede zullen we een student verdeling (T verdeling) met verschillende vrijheidsgraden $\nu$ bekijken. Deze verdeling heeft een kansdichtheidsfunctie van

$$\beta \sim \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})}(1+\frac{\beta^2}{\nu})^{-\frac{\nu+1}{2}}.$$

Voor $\nu = 1$ is deze verdeling gelijk aan de Cauchy verdeling en voor $\nu \to \infty$ is deze verdeling gelijk aan de normale verdeling.

Figuur 4.5 laat zien hoe de student prior eruitziet en hoe deze zich verhoudt tot de Gausische prior. We zien dat de priors op elkaar lijken. De student verdeling heeft echter meer massa bij de uiteinden. Een prior met deze verdeling benadrukt dus nog steeds een bepaalde waarde, maar is gevoeliger voor uitschieters. Op basis van deze eigenschap kunnen we verwachten dat parameters eerder op grotere waarden worden geschat, gezien waarnemingen ver weg van nul nu sterker worden gewogen. We verwachten dus te zien dat puntschattingen voor coëfficiënten met een student prior óf een hele hoge waarde zullen hebben, óf dicht rond nul zullen liggen. Gezien het feit dat de ware data in de simulatie opzet vaak gegenereerd is met coëfficiënten $\beta_j$ die een waarde van nul of één hebben, lijkt de student verdeling een geschikte prior.




Figuur 4.5: De kansdichtheid van de normale verdeling en de student verdeling met verschillende vrijheidsgraden.

## 4.6 Orthogonaliseren

Omdat $X$ gecorreleerd is met hogere orde $X^p$, kan een regressie-analyse $Y = \beta_0 + \beta_1 X + \cdots + \beta_p X^p$ problematisch zijn. De invloed van $X$ op $Y$ wordt dan overschat. Een mogelijke oplossing hiervoor is het orthogonaliseren van de matrix $\mathbf{X}$. Kort gezegd komt dit erop neer dat $X^p$ alleen nog wordt gemodelleerd op het residu wat nog niet is voorspeld door de variabelen $X, \ldots X^{p-1}$. Dit is goed toe te passen in R, door de design matrix van $X$ te vervangen door de georthogonaliseerde design matrix.

## 4.7 Bayesglm()

Voor een regressiemodel $Y = \mathbf{X}\vec{\beta}$ waarbij de $\beta_j$ coëfficiënten onafhankelijke T priors rond $\mu_j$ hebben met schaal $\sigma_j$ kunnen we gebruik maken van het EM algoritme om schattingen voor de coëfficiënten te maken. We maken hiervoor gebruik van de functie *bayesglm()*, een variant op de bekende *general linear model* functies in R. We zullen kort kijken naar de stappen die deze functie uitvoert. De functie maakt afwisselend gebruik van het *iteratively least squares algorithm* waarbij een benadering van het EM algoritme wordt gebruikt. De T prior voor elke coëfficiënt $\beta_j$ wordt als een mengverdeling van

normale verdelingen uitgedrukt met onbekende schaal $\sigma_j$:

$$\beta_j \sim \mathcal{N}(\mu_j, \sigma_j^2), \sigma_j^2 \sim \text{Inv} - \chi^2(\nu_j, s_j^2).$$

Vervolgens wordt er voor elke iteratie gemiddeld over de $\beta_j$'s, waarbij deze als missende data worden gezien. Daarna wordt het EM algoritme toegepast om de $\sigma_j$'s te schatten. Na voldoende iteraties convergeren de coëfficiënten naar de schattingen voor $\beta$.

Het Expectation Maximalisation algoritme bepaalt de verwachting van de log posterior kansdichtheid door een puntschatting te nemen. Vervolgens wordt de verwachte waarde van de log posterior kansdichtheid gemaximaliseerd om een schatting te krijgen voor $\sigma_j^2$. Dit proces van de verwachting schatten en het maximaliseren van de log posterior kansdichtheid wordt herhaald totdat de verwachting convergeert naar een schatting voor $\sigma^2$ (Gelman, 2008).

## 4.8 Onderzoeksvraag

De onderzoeksvraag van dit verslag betreft de vraag wat goede prior keuzes zijn voor een één-dimensionaal polynomiaal Bayesiaans regressiemodel. We zullen verschillende priors bekijken en kijken welke de beste schattingen geven voor het model. De eerste prior die we zullen bekijken en vergelijken is de normale prior met een variabele variantie $\alpha^{-1}$. We zullen hierbij ook kijken naar de schattingen die deze prior geeft wanneer we de matrix $\mathbf{X}$ orthogonaliseren. Daarnaast zullen we kijken naar de schattingen die de T prior geeft, bij deze prior nemen we de vrijheidsgraden $\nu$ variabel. Op basis van de schattingen voor de MSE en de bias van de schattingen die deze priors geven, zullen we kijken welke priors betere schattingen geven voor twee verschillende modellen. In het eerste model simuleren we data van de tweede orde en modelleren we deze ook in een model van orde twee. In het tweede model simuleren we eveneens data van orde twee maar modelleren we deze data in een model van orde vijf. In dit laatste model is er dus sprake van overfitting. We willen onderzoeken hoe de verschillende priors hiermee omgaan. Een goede prior zal aan de ene kant de waarden van de schattingen krimpen waar nodig. Anderzijds moet de prior niet zo conservatief zijn dat alle schattingen een waarde rond nul aannemen, dan worden immers bestaande verbanden ten onrechte onderschat of helemaal niet opgemerkt.

# 5 Simulaties

## 5.1 Het algoritme

We willen de geschiktheid van verschillende soorten priors en priors met verschillende initiële parameters bestuderen. Hiervoor is het volgende experiment opgezet. We genereren allereerst een dataset $(X, Y) = (x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)$. Vervolgens verdelen we onze data in een $K$-aantal partities en passen we kruisvalidatie toe. Voor een bepaalde prior met bepaalde parameterwaarden berekenen we de posterior behorend bij de trainingdata, bestaande uit de data min één partitie. Op basis van deze posterior maken we een puntschatting $\hat{\theta}$. Aan de hand van deze puntschatting voorspellen we de y coördinaten van de testdata uit de weggelaten partitie. Dit proces herhalen we voor elke partitie, zodat we voor elk data punt een ware en een voorspelde waarde hebben. Met deze twee waarden kunnen we nu de MSE uitrekenen. Dit proces herhalen we voor verschillende parameterwaarden van de prior. Na alle verschillende parameters van een prior te hebben doorlopen, kijken we welke parameterwaarde de laagste MSE heeft gegeven. Deze waarde slaan we samen met de bijbehorende MSE op. Het hele proces herhalen we vervolgens een $L$-aantal keer, zodat we een lijst krijgen met duizend 'optimale' parameterwaarden voor een prior. Dit proces herhalen we voor verschillende priors. Het resultaat is een lijst van priors met optimale parameterwaarden en de daarbij behorende MSE.

**for** $l \in \{1 : 1000\}$ **do**
  **Data:** Generate sample $X = (x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)$
  **for** $\alpha_j \in \{\alpha_1, \ldots, \alpha_m\}$ **do**
    **for** $k \in \{1, \ldots, K\}$ **do**
      Using MAP calculate $\vec{\hat{\beta}}_{\alpha_j}$ based on training data;
      Calculate $\hat{y}_k(x, \vec{\hat{\beta}}_{\alpha_j})$ for test data;
    **end**
    $\hat{y} = \left[\hat{y}_1, \ \ldots, \ \hat{y}_k\right]$;
    $\text{MSE}_j = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$;
  **end**
  Save $\alpha_l^* = \alpha_{j'}$ zodanig dat $\text{MSE}_{j'} \leq \text{MSE}_j \forall j \in \{\alpha_1, \ldots, \alpha_m\}$;
  Save $\text{MSE}(\alpha_l^*)$;
**end**

Bovenstaand algoritme is geïmplementeerd in R. De code hiervoor is te vinden in de appendix. Deze code roept de functies aan die het algoritme samen uitvoeren. De functiedefinities zijn te vinden in een aantal externe bestanden die te zien zijn op

*www.github.com/StudentThom/bachelor_project_handin.*

## 5.2 Resultaten

### 5.2.1 Simulatie 1

In de eerste simulatie simuleren we data waarbij $Y$ afhangt van $X$ en $X^2$. We kiezen voor een model van dezelfde vorm. Voor de coëfficiënten $\beta_i$ kiezen we twee verschillende priors. Allereerst nemen we voor beide coëfficiënten een normale verdeling als prior met een variantie $\alpha^{-1}$, waarbij $\alpha \in [0, 10]$. Voor elke coëfficiënt hebben we dus $\beta_i \sim \mathcal{N}(0, \alpha^{-1})$.

Deze prior vergelijken we met dezelfde prior op de georthogonaliseerde dataset $\mathbf{X}$.

We vergelijken bovenstaande priors daarnaast met de student verdeling, waarbij we de vrijheidsgraden variabel kiezen.

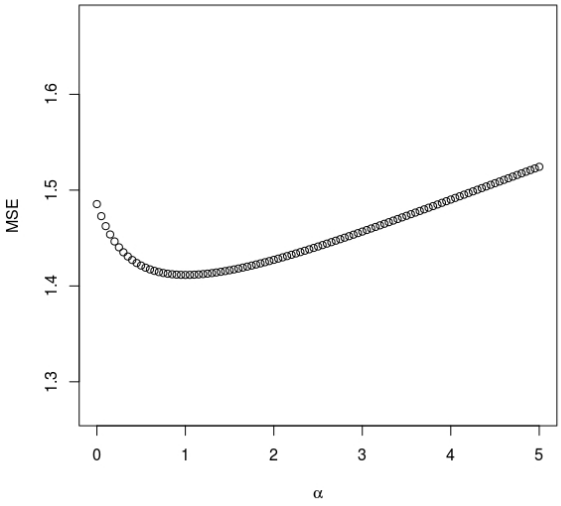
**Simulatie 1**

$$\begin{aligned}&\text{Ware data: } Y = \beta_0 + \beta_1 X + \beta_2 X^2, \text{ waarbij } \beta_0 = \beta_1 = \beta_2 = 1\\ &n = 20\\ &\text{Model } Y = \beta_0 + \beta_1 X + \beta_2 X^2\end{aligned}$$

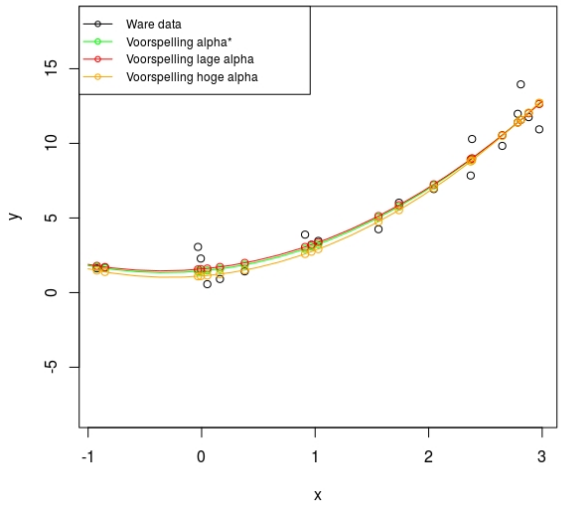
Voor $l = 1$ en met een normale prior rond nul voor elke $\beta$ geeft dit het volgende resultaat:

$$\hat{\beta}_{MAP}(\alpha^*) = \begin{bmatrix}1.44\\0.69\\1.04\end{bmatrix}.$$

We bekijken voor de verschillende varianties de Mean Squared Error, weergeven in figuur 5.1. We zien dat voor lage waarden van $\alpha$ de schattingen een hogere MSE hebben. Naarmate $\alpha$ groter wordt, daalt de MSE. Op een bepaald punt is er een optimum, waar de MSE minimaal is, dit punt geven we aan met $\alpha^*$. Daarna wordt de MSE weer groter naarmate $\alpha$ groter wordt. De interpretatie is dat een Gausische prior met een lage $\alpha$ en dus een grotere variantie, de coëfficiënten te groot schat. Wanneer de variantie iets kleiner wordt, worden de schattingen voor de coëfficiënten lager. Dit geeft kennelijk betere schattingen, want de MSE wordt lager. Wanneer we de variantie echter te klein maken, worden de coëfficiënten te veel naar nul gekrompen. De schattingen zijn dan 'te conservatief' en de MSE wordt weer hoger. Figuur 5.2 geeft de regressielijnen op basis van de MAP schatter met een prior met een lage alfa, een hoge alfa en $\alpha^*$. We zien dat er een verschil is, maar tegelijkertijd is dit verschil relatief klein.




Figuur 5.1: De MSE voor priors met verschillende variantie $\alpha^{-1}$.




Figuur 5.2: De data met de verschillende regressielijnen

Het doel van dit experiment is te kijken naar de afweging tussen het krimpen van schattingen enerzijds en het kijken naar de data anderzijds. We voeren dit experiment daarom 1000 keer uit en bekijken de 1000 verschillende $\alpha^*$ die dit geeft. De frequentie van de verschillende waarden voor $\alpha^*$ is te zien in de histogram in figuur 5.3. Over deze duizend simulaties, elk met haar eigen data en een reeks van priors met variantie

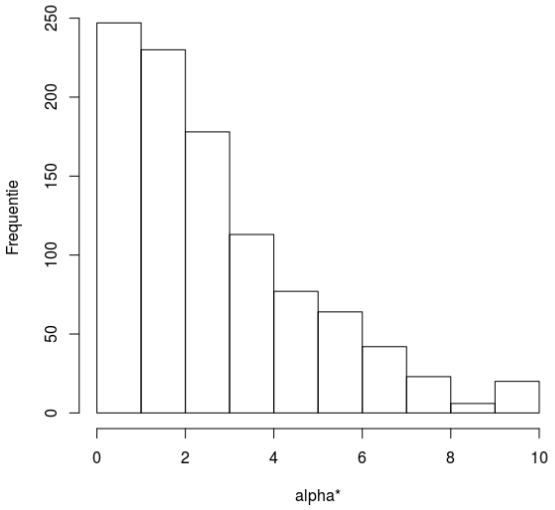
H

Tabel 5.1: Resultaten simulatie 1

| | $\widehat{\text{MSE}}_{\hat{Y}}$ | $\widehat{\text{MSE}}_{\hat{\beta}}$ | $\text{Bias}^2$ |
|---|---|---|---|
| $\mathcal{N}(0, \alpha^*)$ | 1.13 | 1.15 | 0.06 |
| $\mathcal{N}(0, \alpha^*)$, geörthogonaliseerd | 1.22 | - | - |
| T prior | 1.26 | 1.26 | 0.08 |

$\alpha \in [0, 10]$, zien we het volgende gemiddelde van de MAP schatters:

$$\bar{\hat{\beta}}_{MAP} = \begin{bmatrix} 0.85 \\ 0.86 \\ 1.07 \end{bmatrix}.$$




Figuur 5.3: Frequentie van $\alpha^*$ over de duizend simulaties.

De verschillende gemiddelden MSE's van de priors zijn weergeven in tabel 5.1. We zien hier zowel de gekruisvalideerde $\text{MSE}_{\hat{Y}}$ als de test MSE, $\text{MSE}_{\hat{\beta}}$. Deze test MSE is berekend door opnieuw data te genereren met dezelfde coëfficiënten als in de originele samples. De schattingen van de priors zijn vervolgens op deze data getest. Ook de gekwadrateerde bias wordt weergeven. De bias is hier berekend als het gemiddelde verschil van de schattingen van de coëfficiënten en de ware waarden van deze coëfficiënten. Voor de normale prior op de georthogonaliseerde matrix $\mathbf{X}$ is er geen test MSE en geen bias berekend. Zoals gezegd worden bij het berekenen van de test MSE en de bias de schattingen van de coëfficiënten op basis van de prior vergeleken met de ware waarden. Deze schattingen zijn echter ook geschaald door het orthogonaliseren van $\mathbf{X}$, waardoor vergelijken met de ware coëfficiënten lastiger wordt.

De verschillende MSE's van de priors verschillen onderling weinig van elkaar. We zien dat de normale prior met optimale variantie zonder orthogonalisatie de schattingen met de laagste MSE geeft. Voor dit model zou deze prior dus de beste keuze zijn. Merk wel op dat $\alpha^*$ van te voren niet bekend is en moet worden bepaald door voor een groot aantal alfa's de schattingen van de $\mathcal{N}(\mu_0, \alpha^{-1})$ prior te bekijken.

### 5.2.2 Simulatie 2

We hebben nu gezien dat bovenstaande data met behulp van de Bayesiaanse methode goed gemodelleerd kan worden. We zagen echter ook dat er weinig verschil was tussen de MSE van priors met verschillende varianties. In de volgende simulatie gaan we kijken wat er gebeurt als we de data 'overmodelleren'. We nemen nog steeds data samples van de vorm $Y = \beta_0 + \beta_1 X + \beta_2 X^2$. We gebruiken nu echter een vijfde orde model: $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_5 X^5$.

We gebruiken dezelfde priors voor de coëfficiënten $\beta$ als in de eerste simulatie. Merk echter wel op dat we in het nieuwe model zes in plaats van drie coëfficiënten hebben die elk een een eigen, maar wel dezelfde, prior hebben $\beta_i \sim \pi(\theta)$.

**Simulatie 2**

Ware data: $Y = \beta_0 + \beta_1 X + \beta_2 X^2$, waarbij $\beta_0 = \beta_1 = \beta_2 = 1$
$n = 20$
Model: $Y = \beta_0 + \beta_1 X + \cdots + \beta_5 X^5$

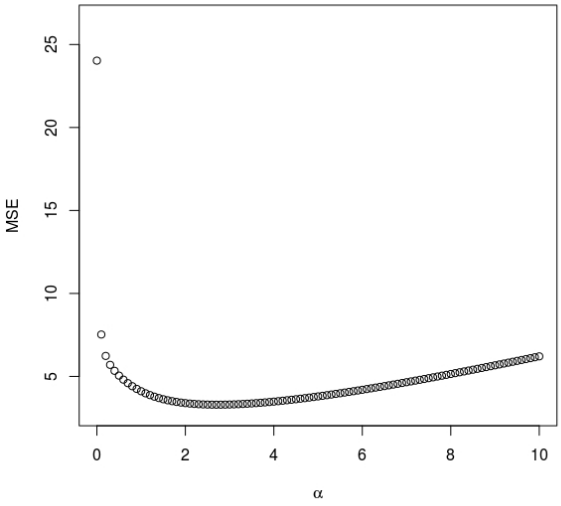
We bespreken de resultaten die we krijgen voor $l = 1$. Figuur 5.4 toont de MSE van de verschillende waarden van $\alpha$. We zien dat de MSE in het algemeen hoger is dan in de vorige simulatie. Bovendien zien we dat het verschil tussen de MSE's voor verschillende alfa's veel groter is. Als we kijken naar de data en de verschillende regressielijnen in figuur 5.5 zien we dat de regressielijn voor een lage alfa ($\alpha = 0.1$) duidelijk van hogere orde is dan orde twee. Er is hier duidelijk sprake van overfitting, zoals besproken in het vorige hoofdstuk. Fouttermen worden niet als zodanig gezien maar worden gemodelleerd door een polynoom van een hoge orde. Per data sample wordt er gebruikt gemaakt van kruisvalidatie. Op basis van de trainingdata wordt een polynoom van hoge orde gemodelleerd. De trainingdata zal een heel lage MSE hebben voor deze regressielijn. De MSE die we zien is echter de MSE van de testdata op basis van de regressielijn van de trainingdata. Deze MSE is dan juist extra hoog doordat de fouttermen 'verkeerd' worden gemodelleerd. In dit geval geldt dat we voor de laagste alfa $\alpha = 0$ de schatting

$$\hat{\beta}_{MAP} = \begin{bmatrix} 2.00 & 1.04 & -1.84 & 1.22 & 0.5 & -0.22 \end{bmatrix}$$

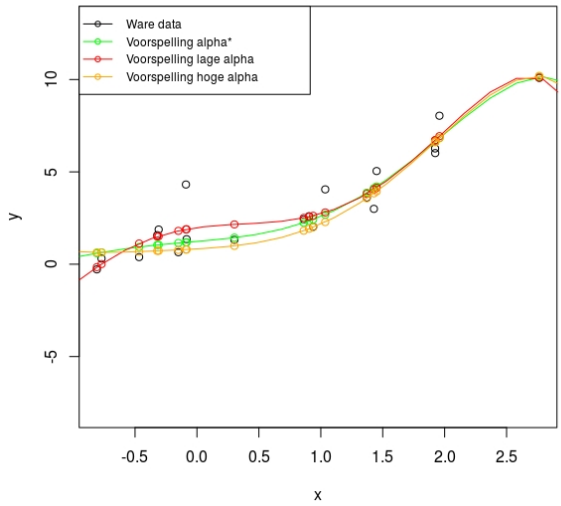
krijgen. Wanneer we echter priors met een wat lagere variantie - en dus een hogere alfa - nemen, zien we dat de schattingen veel beter worden. In dit voorbeeld zien we $\alpha^* = 2.7$, welke de volgende MAP schatters geeft:

$$\hat{\beta}_{MAP} = \begin{bmatrix} 1.18 & 0.53 & 0.13 & 0.60 & 0.20 & -0.11 \end{bmatrix}.$$

De coëfficiënten worden naar nul gekrompen, waardoor het probleem van het overfitten deels wordt voorkomen. Pas voor hele hoge waarden van alfa, als de coëfficiënten te ver naar nul worden gekrompen, wordt de MSE weer hoger.




Figuur 5.4: De MSE voor priors met verschillende variantie $\alpha^{-1}$.




Figuur 5.5: De data met de verschillende regressielijnen

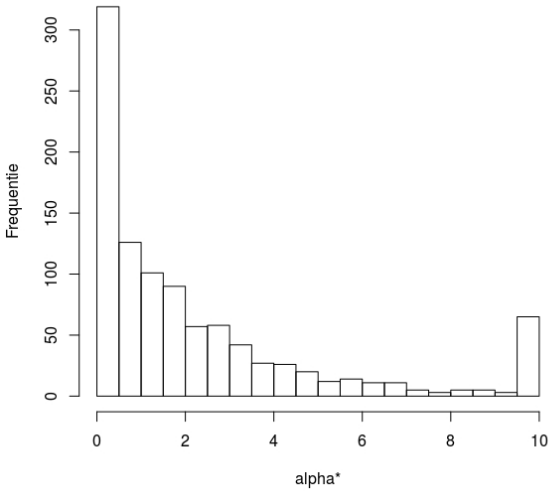
Over de duizend herhalingen zien we dat we gemiddeld de volgende schattingen krijgen

voor de $\mathcal{N}(0, \alpha^*)$ prior:

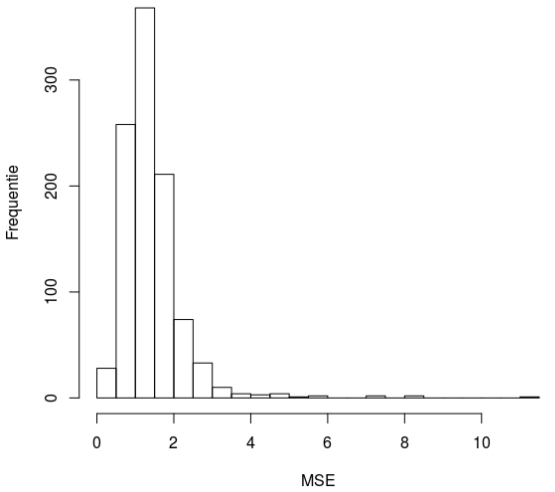
$$\bar{\tilde{\beta}}_{MAP} = \begin{bmatrix} 0.89 & 0.70 & 0.88 & 0.23 & 0.00 & -0.02 \end{bmatrix}$$

, hetgeen redelijk in de buurt komt van de 'ware' waarde

$$\beta = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T.$$




Figuur 5.6: Frequentie van $\alpha^*$ over de duizend simulaties.




Figuur 5.7: MSE's voor van 1000 verschillende $\alpha^*$

H

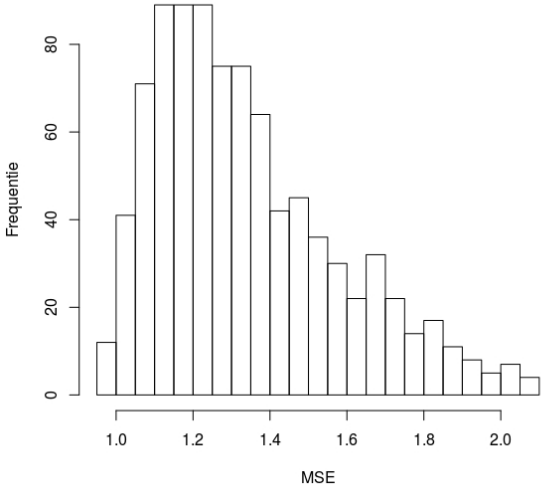
Tabel 5.2: Resultaten simulatie 2

| | $\widehat{\text{MSE}}_{\hat{Y}}$ | $\widehat{\text{MSE}}_{\hat{\beta}}$ | Bias |
|---|---|---|---|
| $\mathcal{N}(0, \alpha^*)$ | 1.48 | 1.81 | 0.54 |
| $\mathcal{N}(0, \alpha^*)$, geörthogonaliseerd | 2.13 | - | - |
| T prior | 4.84 | 6.61 | 1.86 |

H

Tabel 5.3: Resultaten simulatie 1, met getrimd gemiddelde

| | $\widehat{\text{MSE}}_{\hat{Y}}$ | $\widehat{\text{MSE}}_{\hat{Y}}$, trim = 0.1 | $\widehat{\text{MSE}}_{\hat{\beta}}$,$\widehat{\text{MSE}}_{\hat{\beta}}$, trim=0.1 | |
|---|---|---|---|---|
| T prior | 4.84 | 2.44 | 6.61 | 2.15 |




Figuur 5.8: Test MSE's voor van 1000 verschillende $\alpha^*$, hoogste 10% weggelaten.

Over de duizend iteraties zien we in tabel 5.2 dat de schattingen van de MSE's nu verder uit elkaar liggen dan bij de eerste simulatie. We zien opnieuw dat de normale verdeling met variantie $\alpha^*$ de laagste MSE geeft. Dit geldt zowel voor de gekruisvalideerde MSE als de test MSE. Om dezelfde reden als bij simulatie 1 ontbreken de test MSE en de bias voor de prior op de georthogonaliseerde matrix $\mathbf{X}$. We zien ook dat bij de T prior de gemiddelde MSE sterk wordt beïnvloed door een aantal extreem hoge waarden. Het getrimde gemiddelde van de duizend MSE ligt namelijk een stuk lager. Deze MSE heeft echter nog steeds een grotere waarde dan de MSE van de $\mathcal{N}(0, \alpha^*)$ prior.

# 6 Conclusie en discussie

We hebben gezien dat de filosofie achter Bayesiaanse regressiemodellen verschilt van de ideeën achter frequentistische regressie-analyse. De Bayesiaanse methoden geven veel ruimte voor het krimpen van bepaalde parameters. Vergelijkbare technieken bestaan echter ook binnen de frequentistische traditie en zijn in sommige gevallen zelfs equivalent.

Uit de verschillende data simulaties blijkt dat er voor normale priors een optimale variantie is. Deze maakt een afweging tussen de bias en variantie van een schatter, zodat de MSE wordt geminimaliseerd. Deze optimale prior variantie verschilt echter sterk per dataset. Een veilige conservatieve keuze voor de variantie $\alpha^{-1}I$ zou dan een relatief hoge waarde zijn, zodat overfitting zoveel mogelijk wordt voorkomen. Dit is echter ook onbevredigend, omdat mogelijke afhankelijkheden over het hoofd kunnen worden gezien. Een mogelijke tussenweg is dan voor een regressiemodel een hele reeks priors te gebruiken. Vervolgens kan dan de meest geschikte prior worden uitgekozen. Hierbij kan gekeken worden naar de MSE, maar ook de subjectiviteit van de statisticus kan een rol spelen. Belangrijk is echter in de gaten te houden dat een model niet afhangt van een specifiek data sample. Een schatting voor regressiecoëfficiënten moet zo gemaakt worden dat deze ook voor een nieuw sample een goede fit geeft. Op die manier blijft de analyse reproduceerbaar.

In het geval waar we data van de tweede orde hadden en we deze data ook in een tweede orde model modelleerden, zagen we dat een normale prior rond nul met een variantie van $\alpha^{*-1}$ de beste schattingen gaf voor het model. Het verschil met dezelfde prior maar met georthogonaliseerde onafhankelijke variabelen was echter klein. Ook de T prior met optimale vrijheidsgraden $\nu^*$ gaf een vergelijkbare MSE die relatief niet veel groter was. Het verschil tussen deze verschillende priors werd echter veel groter wanneer data van dezelfde orde werd gemodelleerd in een model waarbij wordt aangenomen dat $Y$ van $X+X^2+\cdots+X^5$ afhangt. We zagen in dat geval dat de MSE van de schattingen duidelijk lager is bij de normale prior met optimale variantie dan de MSE van de schattingen van zowel normale priors met een andere variantie als de T prior. Ook het orthogonaliseren van $\mathbf{X}$ leverde geen verbetering op in de MSE van de schattingen.

De normale prior met optimale variantie maakt per definitie de afweging tussen de bias en de variantie zodanig dat de MSE geminimaliseerd wordt. Dit verklaart waarom deze prior in de simulaties in vergelijking met andere priors lagere MSE's geeft. De normale prior met een hele hoge variantie maakt bijna dezelfde schattingen als de OLS schatter. Deze schattingen hebben weinig bias maar hebben een grotere variantie. De MSE van deze schatters is daardoor groter dan de MSE van priors met een lagere variantie.

We hebben in de simulaties nu alleen naar normale priors met $\mu_0 = 0$ gekeken en niet naar normale priors met een andere $\mu_0$. Wanneer men van te voren een idee heeft over de waarschijnlijke waarden van te schatten coëfficiënten, zou men de normale priors

voor deze coëfficiënten kunnen centreren rond deze waarden. De schattingen zullen dan waarschijnlijk beter worden, maar het gevaar is dat wanneer het vermoeden vooraf ten onrechte blijkt te zijn, dit minder snel door het statistische model gezien wordt.

Een andere mogelijkheid voor het kiezen van de parameterwaarden voor de prior zou dan Emperical Bayes zijn (Gelman et al, 2014). Hierbij worden schattingen voor de coëfficiënten van de priors gemaakt op basis van de data. De data wordt dan twee keer gebruikt, een keer voor het schatten van de parameters van de priors en een keer voor de statistische analyse. Men kan echter ook de data opsplitsen en een deel van de data voor het schatten van priorparamaters gebruiken. Het andere deel van de data kan dan gebruikt worden voor de Bayesiaanse regressie-analyse met de prior met de geschatte initiële parameterwaarden.

Nog een andere mogelijkheid voor het kiezen van de parameters van de prior is het gebruik maken van een hyperprior. Hierbij krijgt de parameter van de prior een eigen priorverdeling. Dit kan bijvoorbeeld een gammaverdeling zijn met paramters $\alpha$ en $\beta$. Deze parameters kunnen dan vrij neutraal gekozen worden. Deze techniek is al kort besproken in het hoofdstuk over Bayesiaanse regressie. Met deze techniek hoeft er geen hele reeks van parameterwaarden voor de prior uitgeprobeerd te worden, waardoor mogelijk minder berekeningen vereist zijn. Bovendien heeft dit model ook een duidelijke interpretatie.

# Bibliografie


[1] Bishop, C.M., *Pattern Recognition and Machine Learing* Springer, 2006.

[2] Gelman, A., Jakulin, A., Grazia Pittau, A. & Yu-Sung Su, *A weakly informative deafualt prior distribution for logistic and other regression models*, The Annal of Applied Statistics, **Vol. No. 4**, 2008.

[3] Gelman, A. et al., *Bayesian Data Analysis*, CRC Press, Third Edition, 2014.

[4] Hastie, T., Tibshirani, R., Friedman, J., *The Elements of Statistical Learning*, Springer, 2001.

[5] Kleijn, B.J.K., *The frequentist theory of Bayesian statistics*, Springer, 2018.

[6] Murphy, K.M., *Machine Learing: A Probabilistic Perspective*, MIT Press, 2012.

[7] Wieringen, W.N. van, *Lecture notes on ridge regression*, 2018.

# 7 Appendix

```
##############################################
#
# execute exp2_bayes_norm_priors.R
#
##############################################

setwd(paste("Documents/wiskunde/",
            "2017-2018/bachelor_project/R/handin",sep=""))
setwd(paste("/media/mynewdrive1/Documenten/Wiskunde/2017-2018/",
            "bachelor_project/R/handin",sep=""))

# libraries
library("arm")
# load functions
source("helpers.R")
source("helpers_CV_bayesglm.R")
# load variables
source("variables_exp2.R")

# Choose what initial betas and sample size to use
## Betas
k = 1
## Sample size
l = 1

# for testing
repeat_cross_val <- 1001
alpha <- seq(0.001,15,length=101)
order_fit <- 2
alpha_plot = TRUE; data_plot = TRUE
alpha_plot = FALSE; data_plot = FALSE
orthog <- TRUE
orthog <- FALSE

# Execute experiment
alpha_stars <- c()
```

```
mses <- c()
mu_news <- c()
for (i in 1:repeat_cross_val){
  cross_val_temp <- execute_cross_val(alpha = alpha, x_min = x_min,
                                      x_max = x_max,
                                      order_sample = order_p,
                                      order_fit = order_fit,
                                      initial_betas = initial_betas[[k]],
                                      sample_size = sample_size[l],
                                      orthog =orthog, alpha_plot =
alpha_plot,
                                      data_plot = data_plot,
                                      multiple_samples = FALSE)
  #
  alpha_stars[i] <- cross_val_temp[[1]]$alpha_star
  mses[i] <- cross_val_temp[[1]]$alpha_star_cross_val$mse
  mu_news[[i]] <- cross_val_temp[[1]]$alpha_star_cross_val$mu_new
}

# calc out of back mse = test mse using mu_news
i <- 1
test_mses <- c()
for (beta in mu_news[1:1001]) {
  test_mses[i] <- out_of_back_mse(beta_hat = beta, beta_true = c(1,1,1),
                                  x_min = x_min, x_max = x_max, n = 1000)
  i <- i + 1
}
mean(test_mses)

# Visualise experiment
## alphas
hist(alpha_stars,xlab=paste(expression(alpha),"*",sep=''),
     ylab="Frequentie",main="")
plot(alpha_stars,mses,xlab=paste(expression(alpha),"*",sep=''),
     ylim=c(0,13))
plot(alpha_stars)
## (test) mse's
hist(mses,breaks =20,main="",xlab="MSE",ylab="Frequentie")
hist(test_mses,main="",xlab="MSE",ylab="Frequentie")
hist(sort(test_mses)[1:(0.9*length(test_mses))],breaks=20,main="",
     xlab="MSE",ylab="Frequentie")


# numbers
```

```
mean(alpha_stars)
mean(mses)
mu_1 <- 0
mu_2 <- 0
mu_3 <- 0
mu_4 <- 0
mu_5 <- 0
mu_6 <- 0
for (j in 1:length(mu_news)){
  mu_1 <- mu_1 + mu_news[[j]][1]
  mu_2 <- mu_2 + mu_news[[j]][2]
  mu_3 <- mu_3 + mu_news[[j]][3]
  mu_4 <- mu_4 + mu_news[[j]][4]
  mu_5 <- mu_5 + mu_news[[j]][5]
  mu_6 <- mu_6 + mu_news[[j]][6]
}
mu_1_mean <- mu_1 / length(mu_news)
mu_2_mean <- mu_2 / length(mu_news)
mu_3_mean <- mu_3 / length(mu_news)
mu_4_mean <- mu_4 / length(mu_news)
mu_5_mean <- mu_5 / length(mu_news)
mu_6_mean <- mu_6 / length(mu_news)

##############################################
#
# execute exp3_bayes_t_priors.R
#
##############################################

setwd("Documents/wiskunde/2017-2018/bachelor_project/R/handin")
setwd(paste("/media/mynewdrive1/Documenten/Wiskunde/2017-2018/",
            "bachelor_project/R/handin",sep=""))
# libraries
library("arm")
# load functions
source("helpers.R")
source("helpers_CV_bayesglm.R")
# load variables
source("variables_exp3.R")

# Choose what initial betas and sample size to use
## Betas
k = 1
initial_betas[[k]] = c(1,1,1,1,1,1)
```

```
## Sample size
l = 1
order_fit <- 2
repeat_cross_val <-1001
df_plot =  TRUE; data_plot = TRUE
df_plot =  FALSE; data_plot = FALSE

# Execute experiment
for (i in 1:repeat_cross_val){
  cross_val_temp <- execute_cross_val_tprior(df = df, x_min = x_min,
                                             x_max = x_max,
                                             order_sample = order_p,
                                             order_fit = order_fit,
                                             initial_betas = initial_betas
                                             sample_size = sample_size[l],
                                             orthog =FALSE, df_plot =
df_plot,
                                             data_plot = data_plot,
                                             multiple_samples = FALSE)
  #
  df_stars[i] <- cross_val_temp[[1]]$df_star
  mses[i] <- cross_val_temp[[1]]$df_star_cross_val$mse
  mu_news[[i]] <- cross_val_temp[[1]]$df_star_cross_val$mu_new
  print(i)
}

# calc out of back mse = test mse using mu_news
i <- 1
test_mses <- c()
for (beta in mu_news[1:1001]) {
  #print(beta)
  test_mses[i] <- out_of_back_mse(beta_hat = beta, beta_true = c(1,1,1),
                                  x_min = x_min, x_max = x_max, n = 100)
  i <- i + 1
}
mean(test_mses)
mean(test_mses,trim=0.1)
hist(test_mses)
hist(test_mses,breaks=10000,main="",xlab="MSE",ylab="Frequentie")
hist(sort(test_mses)[1:(0.8*length(test_mses))],breaks=10,main="",
     xlab="MSE",ylab="Frequentie")

# Visualise experiment
hist(df_stars,main="",xlab="df",ylab="Frequentie")
```

```
plot(df_stars,mses)
plot(df_stars)
hist(mses,breaks =50,main="",xlab="MSE",ylab="Frequentie")
# numbers
mean(df_stars)
mean(mses)
mean(mses,trim=0.1)
mu_1 <- 0
mu_2 <- 0
mu_3 <- 0
mu_4 <- 0
mu_5 <- 0
mu_6 <- 0
for (j in 1:length(mu_news)){
  mu_1 <- mu_1 + mu_news[[j]][1]
  mu_2 <- mu_2 + mu_news[[j]][2]
  mu_3 <- mu_3 + mu_news[[j]][3]
  mu_4 <- mu_4 + mu_news[[j]][4]
  mu_5 <- mu_5 + mu_news[[j]][5]
  mu_6 <- mu_6 + mu_news[[j]][6]
}
mu_1_mean <- mu_1 / length(mu_news)
mu_2_mean <- mu_2 / length(mu_news)
mu_3_mean <- mu_3 / length(mu_news)
mu_4_mean <- mu_4 / length(mu_news)
mu_5_mean <- mu_5 / length(mu_news)
mu_6_mean <- mu_6 / length(mu_news)


mu_news_mean <- c(mu_1_mean,mu_2_mean,mu_3_mean,
                  mu_4_mean,mu_5_mean,mu_6_mean)
# bias
verschil1 <- abs(c(1,1,1,1,1,1)-c(mu_1_mean,mu_2_mean,mu_3_mean,
                                  mu_4_mean,mu_5_mean,mu_6_mean))
bias_sq <- mean(verschil1)**2
bias_sq

verschil2 <- 0
for (beta in mu_news[1:1000]){
  verschil2 <- verschil2 + abs(c(beta[1],beta[2],beta[3]) - c(1,1,1))
}
bias_sq2 <- mean(verschil2 / 1000)**2
bias_sq2
```