Modellenpracticum verslag

Alex van der Hulst Marten Straatsma Michiel Barten
Sièna van Schaick Simcha van Collem Thomas Berghuis

1 juli 2022

Inhoudsopgave

1 Introductie

Zelfrijdende auto's moeten om kunnen gaan met allerlei verschillende verkeerssituaties. Een manier om te bepalen welke verkeerssituaties voorkomen, is door data uit echte verkeerssituaties te verzamelen, en deze te beschrijven met behulp van statistische modellen. Er zijn meerdere manieren om data te verzamelen die verschillende verdelingen geven. Hoe kunnen we datasets met verschillende verdelingen toch combineren om een idee te krijgen van de gehele situatie?

Het geval dat wij in dit verslag beschouwen is dat de verschillende manieren om data te verzamelen het continu verzamelen van normale data tegenover het verzamelen van extra data in interessante gevallen zijn. De data verkregen uit de interessante gevallen noemen we 'edge case data'.

De data bestaat bijvoorbeeld uit de snelheid en de huidige remkracht van de auto, waarbij dit elke 10 seconden wordt gemeten. Voor de normale data wordt dit slechts 1 keer per uur opgeslagen. Echter als er een erg hoge remkracht is zullen alle auto's elke 10 seconden deze data opslaan. De erg hoge remkracht is dan het 'edge case criterium', dat aangeeft dat er een interessante situatie speelt. De vraag die we dan mogelijk willen beantwoorden is hoe we alle data kunnen gebruiken om de verdeling van de snelheid van auto's te bepalen. Uiteraard kan dit door alleen gebruik te maken van de normale data. Echter zal hierin de schatting voor de verdeling rondom extreem hoge snelheden niet precies zijn, omdat hier weinig data van is. Dit kan mogelijk verbeterd worden met behulp van de edge case data als er een positieve correlatie is tussen hoge remkracht en hoge snelheden.

In dit verslag zullen wij methodes ontwikkelen die in het geval van het voorbeeld kunnen helpen de verschillende datasets te combineren. Het voorbeeld is leidend geweest in de keuze van het specifiekere wiskundige probleem dat we op gaan lossen, dat afgeleid is van een algemener wiskundig probleem over het combineren van verschillend verdeelde datasets. Dit wordt toegelicht in de volgende paragraaf.

We zullen allereerst deze vrij informele beschrijving van het probleem formaliseren door middel van theorie uit de statistiek. Daarna volgt een hoofdstuk met theorie dat we gebruiken voor het onderbouwen van de ontwikkelde methodes en het testen van de methodes. Vervolgens lichten we van drie verschillende methodes toe wat het idee ervan is, hoe het theoretisch in elkaar steekt en hoe het geïmplementeerd kan worden. Daarna testen we de methodes en worden ze met elkaar vergeleken. Tot slot trekken we hieruit conclusies over onze methodes en reflecteren wij op dit onderzoek in de discussie.

Om onze methodes te kunnen testen, genereren we in onze implementaties van de methodes data waarop we vervolgens de methodes hebben toegepast. Dit zal ook nog beschreven worden in de uitleg van de implementaties van de methodes. Aanvankelijk hebben wij ons in onze testen beperkt tot positief gecorreleerde, tweedimensionale, normaal verdeelde data die we genereerden met behulp van Pythons `SciPy` library. Uiteindelijk hebben wij dit uitgebreid naar willekeurig gecorreleerde, tweedimensionale data getrokken uit diverse verdelingen die we genereerden met behulp van de `lcmix` library in R.

1.1 In wiskundige termen

Van alle data valt er dus een deel onder zogeheten edge case data. Wat egde case data is, wordt bepaald door één of ander criterium dat in principe van alles kan zijn. Voorbeelden die aansluiten bij de introductie zijn data waarbij de remkracht van een bepaalde grootte is of data vanaf een bepaalde snelheid. Van deze edge case data wordt alles opgeslagen, terwijl van de normale data het grootste deel weg wordt gegooid. Hierbij weten we nog wel hoeveel data er is weggegooid, dus de verhouding tussen normale situaties en situaties passend bij de edge case data is bekend.

Wiskundig kunnen we dit dus modelleren door data te trekken uit één of andere verdeling (mogelijk van meerdere dimensies) en een edge case criterium te bepalen. We nemen hierbij niets aan over met welk type verdeling we te maken hebben. Een klein percentage van deze data houden we apart. Dit is dan de normale data. Merk op dat een deel van de normale data ook aan het edge case criterium kan voldoen. Verder behouden we van alle data ook de data die aan het edge case criterium voldoet. Dit wordt dan de edge case data. Deze data kan deels overlappen met de normale data. De rest van de data gebruiken we niet meer. Wel weten we hoeveel data we weg hebben gegooid. Nu is de vraag of en in hoeverre we de normale data samen met de edge case data kunnen gebruiken om de verdeling te schatten van (een deel van) de originele verdeling. Omdat we mogelijk data trekken uit een hoger-dimensionale verdeling, kan het zijn dat we alleen op zoek zijn naar de verdeling van één van de componenten.

Een vergelijkbaar statistisch probleem is waarbij je gegeven een set data getrokken uit een onbekende verdeling, de verdeling moet schatten. Het verschil met het hier gepresenteerde probleem is dat de data waarop de schatter van de verdeling gebaseerd moet worden als het ware scheef verdeeld is. Data dat voldoet aan het edge case criterium is oververtegenwoordigd. Dit moet op één of andere manier weer gecompenseerd worden.

Om het probleem wat concreter en van de goede omvang te maken, zullen we in dit verslag alleen naar een wat specifieker geval kijken. We zullen het geval beschouwen waarin we data trekken uit een tweedimensionale verdeling. We krijgen dan data van de vorm $\{(X_i, Y_i)\}_i$, waarbij $\forall_{i \leq n}\ X_i, Y_i \in \mathbb{R}$. Het edge case criterium is dat de tweede component groter is dan één of andere constante $c \in \mathbb{R}$ en de verdeling waar we in geïnteresseerd zijn, is die van de eerste component. In hoeverre de methodes die we ontwikkelen in dit verslag toepasbaar zijn in het algemene geval, zullen we bespreken in de conclusie.

We voeren eerst wat notatie in die we de rest van dit verslag aanhouden. Daarna beschrijven we het probleem op een precieze wiskundige manier met behulp van deze notatie.

1.1.1 Overzicht notatie

Er volgt nu een opsomming van de notatie die we gebruiken in dit verslag.

- X en Y zijn continue reëelwaardige stochasten. X is hierbij de coördinaat waarvan we geïnteresseerd zijn in de verdeling. De waarde van Y bepaalt wanneer we spreken over edge case data;
- $f_{X,Y} : \mathbb{R}^2 \to \mathbb{R}$ is de tweedimensionale kansdichtheidsfunctie van X en Y samen;
- $f_X, f_Y : \mathbb{R} \to \mathbb{R}$ zijn respectievelijk de kansdichtheidsfunctie van X en van Y, deze zijn vastgelegd door $f_{X,Y}$ te integreren over Y en X;
- $\hat{f}_{X,Y}$, $\hat{f}_X$ en $\hat{f}_Y$ zijn schatters van de bovenstaande kansdichtheidsfuncties. Uit de context moet blijken op welke manier deze bepaald zijn;
- $F_X, F_Y : \mathbb{R} \to [0, 1]$ zijn respectievelijk de verdelingsfuncties van X en van Y;
- $\hat{F}_X$ en $\hat{F}_Y$ zijn schatters van de bovenstaande verdelingsfuncties. Uit de context moet blijken op welke manier deze bepaald zijn;
- $f_{X|Y>c} : \mathbb{R} \to \mathbb{R}$ is de kansdichtheidsfunctie van de voorwaardelijke verdeling van X gegeven dat $Y > c$;
- $f_{X,Y|Y>c} : \mathbb{R}^2 \to \mathbb{R}$ is de kansdichtheidsfunctie van de voorwaardelijke verdeling van X en Y samen gegeven dat $Y > c$.

1.1.2 Het probleem in wiskundige termen

In wiskundige termen zou ons probleem er dus als volgt uit zien. We beginnen met het trekken van $N + M$ datapunten uit de verdeling met kansdichtheid $f_{X,Y}$. Vervolgens behouden we alleen de volgende gegevens:

- $c \in \mathbb{R}$: Een constante die aangeeft wanneer bepaalde data telt als edge case data;
- $\begin{pmatrix} X_i \\ Y_i \end{pmatrix}$ voor $i \in \{1, \ldots, K\}$, een kleine hoeveelheid normale data. Dit is dus gewoon een deel van alle data getrokken uit de verdeling met kansdichtheid $f_{X,Y}$;
- $\begin{pmatrix} X_i \\ Y_i \end{pmatrix}$ voor alle $i \in \{K + 1, \ldots, K + N\}$, de N datapunten getrokken uit de verdeling met kansdichtheid $f_{X,Y}$ waarbij $Y > c$. Dit is dus de edge case data en verdeeld met kansdichtheid $f_{X,Y|Y>c}$;
- M: het aantal datapunten met $Y \leq c$.

En te bepalen is:

- $\hat{f}_X$: een schatter voor de kansdichtsheidsfunctie van X.

2 Voorkennis

In dit hoofdstuk behandelen we de theorie die onze methodes ondersteunen. In bijzonder werken we veel met kernschatters. Aangezien we niet aannemen dat X één of andere verdeling heeft (zoals een normale verdeling of een exponentiële verdeling), kunnen we de verdeling niet schatten door parameters te schatten. Kernschatters zijn precies in deze situatie handig. Ze geven een methode om een kansdichtheidsfunctie te schatten zonder aannames over een verdeling te gebruiken. Verder lichten we kort toe hoe tweedimensionale kansdichtheidsfuncties werken, wat de law of total probability is en wat het betekent om te conditioneren met kans 0. Tot slot gaan we in op theorie die we kunnen gebruiken om de correctheid te meten van een schatter, zodat we later verschillende methodes met elkaar kunnen vergelijken.

2.1 Tweedimensionale kansdichtheidsfuncties

De verdeling van een continue reële stochast X kun je beschrijven met behulp van een kansdichtheidsfunctie f_X. De kans op een bepaalde gebeurtenis $D \subseteq \mathbb{R}$ is dan gelijk aan $\mathbb{P}[X \in D] = \int_D f_X(x)dx$. Dit idee kunnen we generaliseren naar gecombineerde verdelingen met vectoren van stochasten. We beschouwen alleen het geval met twee stochasten, omdat dit als enige relevant is voor ons. De kansdichtheidsfunctie $f_{X,Y} : \mathbb{R}^2 \to \mathbb{R}$ van de verdeling van (X, Y) werkt dan eigenlijk precies hetzelfde. De kans op een bepaalde gebeurtenis $D \subseteq \mathbb{R}^2$ is gelijk aan $\mathbb{P}[(X, Y) \in D] = \int_D f_{X,Y}(x, y)dxdy$.

2.2 The law of total probability

De law of total probability is een bekende identiteit uit de kansrekening die het soms makkelijker maakt om kansen uit te rekenen. Deze zullen we hier intuïtief toelichten, maar niet formeel bewijzen. Het maakt gebruik van conditionele kansen via de formule $\mathbb{P}[A, B] = \mathbb{P}[A|B] \cdot \mathbb{P}[B]$. De law of total probability komt eigenlijk neer op een algemenere variant van $\mathbb{P}[A] = \mathbb{P}[A, B] + \mathbb{P}[A, B^c] = \mathbb{P}[A|B] \cdot \mathbb{P}[B] + \mathbb{P}[A|B^c] \cdot \mathbb{P}[B^c]$. Hier splitsen we de gebeurtenis A op in twee gevallen, namelijk dat de gebeurtenis B plaatsvindt en dat de gebeurtenis B niet plaatsvindt. We kunnen ook voor een discrete stochast Y die bijvoorbeeld waarden aanneemt in $\mathbb{N}$ de gebeurtenis A opsplitsen in alle mogelijke gevallen van waarden van Y. Dit geeft dan op dezelfde manier de volgende formule:

$$\mathbb{P}[A] = \sum_n \mathbb{P}[A|Y = n] \cdot \mathbb{P}[Y = n]$$

Dit is dan ook de total law of probability in het discrete geval. Wij gebruiken de continue variant van de law of total probability, waarin Y een continue stochast is. De formule lijkt erg op die van het discrete geval, maar in het continue geval kunnen we niet meer alle gevallen $Y = y$ voor alle $y \in \mathbb{R}$ bij elkaar op tellen, omdat dit overaftelbaar veel gevallen zijn. Daarom krijgen we in dit geval een integraal. Voor een continue stochast Y geldt verder dat $P[Y = y] = 0$ voor alle $y \in \mathbb{R}$, dus kijken we in plaats daarvan naar de waarde van de kansdichtheidsfunctie $f_Y(y)$. Al met al is de total law of probability in het continue geval het volgende:

$$\mathbb{P}[A] = \int_{-\infty}^{\infty} \mathbb{P}[A \mid Y = y] \cdot f_Y(y)\, dy$$

Hoewel het intuïtief duidelijk is wat $\mathbb{P}[A \mid Y = y]$ betekent (stel dat $Y = y$, wat is dan de kans op A?), moeten we volgens de definitie delen door 0 om dit te berekenen: $\mathbb{P}[A \mid Y = y] = \frac{\mathbb{P}[A, Y=y]}{\mathbb{P}[Y=y]}$. Hoe we soms toch met dit soort uitdrukkingen kunnen rekenen, wordt toegelicht in de volgende paragraaf.

2.3 Conditioneren met kans 0

We willen termen van de vorm $\mathbb{P}[A \mid B]$ bekijken, waarbij $\mathbb{P}[B] = 0$. Dit lijkt problematisch als we naar de definitie van voorwaardelijke kans kijken:

$$\mathbb{P}[A \mid B] = \frac{\mathbb{P}[A, B]}{\mathbb{P}[B]}$$

Dit probleem kan echter voorkomen worden in sommige situaties. In het volgende voorbeeld zullen we $\mathbb{P}[X \leq x_0 \mid Y = y_0]$ bespreken aangezien we deze later nodig hebben, maar deze methode kan ook voor andere voorwaardelijke kansen gebruikt worden.

Zij X, Y continu reële simultaan verdeelde stochasten met kansdichtheidsfunctie $f_{X,Y}(x, y)$. Neem $x_0, y_0 \in \mathbb{R}$ en zij $\varepsilon > 0$. We zetten $A = \{X \leq x_0\}$ en $B = \{|Y - y_0| < \varepsilon\}$. Dat betekent dat

$$\mathbb{P}[X \leq x_0 \mid |Y - y_0| < \varepsilon] = \frac{\mathbb{P}[X \leq x_0, |Y - y_0| < \varepsilon]}{\mathbb{P}[|Y - y_0| < \varepsilon]} = \frac{\int_{y_0-\varepsilon}^{y_0+\varepsilon} \int_{-\infty}^{x_0} f_{X,Y}(x, y) dx dy}{\int_{y_0-\varepsilon}^{y_0+\varepsilon} \int_{-\infty}^{\infty} f_{X,Y}(x, y) dx dy}$$

Nu kunnen we $\mathbb{P}[X \leq x_0 \mid Y = y_0]$ definiëren als het limiet van deze uitdrukking als ε naar 0 gaat.

$$\mathbb{P}[X \leq x_0 \mid Y = y_0] = \lim_{\varepsilon \to 0} \frac{\int_{y_0-\varepsilon}^{y_0+\varepsilon} \int_{-\infty}^{x_0} f_{X,Y}(x, y) dx dy}{\int_{y_0-\varepsilon}^{y_0+\varepsilon} \int_{-\infty}^{\infty} f_{X,Y}(x, y) dx dy}$$

Aangezien zowel de teller als de noemer naar 0 gaan, kunnen we de regel van l'Hôpital gebruiken.

$$\mathbb{P}[X \leq x_0 \mid Y = y_0] = \lim_{\varepsilon \to 0} \frac{\frac{d}{d\varepsilon} \int_{y_0-\varepsilon}^{y_0+\varepsilon} \int_{-\infty}^{x_0} f_{X,Y}(x, y) dx dy}{\frac{d}{d\varepsilon} \int_{y_0-\varepsilon}^{y_0+\varepsilon} \int_{-\infty}^{\infty} f_{X,Y}(x, y) dx dy}$$

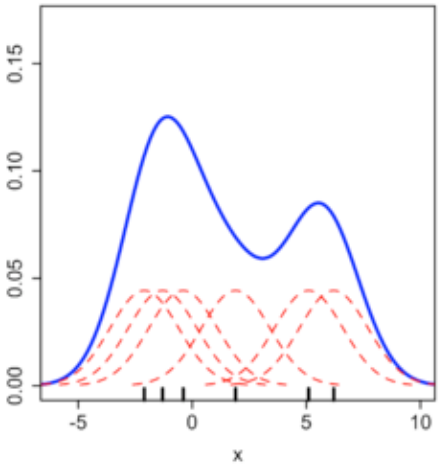
Uit de integraalregel van Leibniz, een generalisatie van de hoofdstelling van de calculus, volgt dan dat

$$\begin{aligned}
&\mathbb{P}[X \leq x_0 \mid Y = y_0] \\
&= \lim_{\varepsilon \to 0} \frac{\int_{-\infty}^{x_0} f_{X,Y}(x, y_0 + \varepsilon) dx + \int_{-\infty}^{x_0} f_{X,Y}(x, y_0 - \varepsilon) dx + \int_{y_0+\varepsilon}^{y_0+\varepsilon} \left(\frac{\partial}{\partial \varepsilon} \int_{-\infty}^{x_0} f_{X,Y}(x, y) dx \right) dy}{\int_{-\infty}^{\infty} f_{X,Y}(x, y_0 + \varepsilon) dx + \int_{-\infty}^{\infty} f_{X,Y}(x, y_0 - \varepsilon) dx + \int_{y_0+\varepsilon}^{y_0+\varepsilon} \left(\frac{\partial}{\partial \varepsilon} \int_{-\infty}^{\infty} f_{X,Y}(x, y) dx \right) dy} \\
&= \frac{2 \cdot \int_{-\infty}^{x_0} f_{X,Y}(x, y_0) dx}{2 \cdot \int_{-\infty}^{\infty} f_{X,Y}(x, y_0) dx} \\
&= \frac{\int_{-\infty}^{x_0} f_{X,Y}(x, y_0) dx}{\int_{-\infty}^{\infty} f_{X,Y}(x, y_0) dx}
\end{aligned}$$

2.4 Kernschatters

Voor methodes die we gebruiken, maken we gebruik van een kernschatter. Dit is een (meestal) continue functie die de kansdichtheidsfunctie probeert te benaderen door over elk datapunt een kernfunctie K te zetten, deze bij elkaar op te tellen en vervolgens te normaliseren. Over het algemeen wil men dat een kernfunctie integreert naar 1, niet-negatief, symmetrisch en dalend na 0 is:

$$\begin{aligned}
\text{integreert naar 1:} \quad & \int_{-\infty}^{\infty} K(x) dx = 1 \\
\text{niet-negatief:} \quad & K(x) \geq 0 \\
\text{symmetrisch:} \quad & K(x) = K(-x) \\
\text{dalend na 0:} \quad & K'(x) \leq 0 \ \forall x > 0
\end{aligned}$$



Figuur 1: Voorbeeld Kernschatter [5]

De eerste twee eigenschappen zijn gewenst omdat we met een kansdichtheid te maken hebben. De symmetrie is gewenst omdat men aanneemt dat de kansen links en rechts van dat punt gelijk zijn. Dat de functie dalend na 0 is, is gewenst omdat men aanneemt dat de kans verder van het punt vandaan, kleiner zou moeten zijn.

Een kernfunctie kan bijvoorbeeld een Gausscurve zijn. Een kernschatter van data $(x_1, x_2, \dots x_n)$ wordt formeel gegeven door:

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

De letter h staat hier voor de bandbreedte die nog gekozen kan worden. In het geval dat de kernfuncties Gaussecurves zijn, bepaalt h de standaarddeviatie. Een te grote h zorgt ervoor dat de pieken te laag en glad zijn en een te kleine h voor te steile pieken. Deze h wordt dus gekozen op basis van de spreiding en omvang van de dataset.

2.4.1 Multidimensionale kernschatters

Het idee van deze kernschatter kunnen we uitbreiden naar meerdere dimensies. Wederom willen we doorgaans een kernfunctie die voldoet aan de volgende eisen:

$$\begin{aligned}
\text{integreert naar 1:} &\quad \int_{\mathbb{R}^d} K(x)dx = 1 \\
\text{niet-negatief:} &\quad K(x) \geq 0 \\
\text{symmetrisch:} &\quad K(x) = K(-x) \\
\text{dalend na 0:} &\quad \frac{d}{dx_i}K(x) \leq 0 \ \forall x \in (\mathbb{R}_{>0})^d
\end{aligned}$$

In dit verslag beperken we ons tot de tweedimensionale kernschatter. De kernschatter van data $\left(\begin{pmatrix}x_1\\y_1\end{pmatrix}, \begin{pmatrix}x_2\\y_2\end{pmatrix}, \dots \begin{pmatrix}x_n\\y_n\end{pmatrix}\right)$ wordt dan formeel gegeven door:

$$\hat{f_H}(x, y) = \frac{1}{n}\sum_{i=1}^{n} K_H\left(\begin{pmatrix}x\\y\end{pmatrix} - \begin{pmatrix}x_i\\y_i\end{pmatrix}\right) = \frac{1}{n \cdot \sqrt{\det(H)}}\sum_{i=1}^{n} K\left(\frac{1}{\sqrt{H}}\left(\begin{pmatrix}x\\y\end{pmatrix} - \begin{pmatrix}x_i\\y_i\end{pmatrix}\right)\right)$$

De letter H staat hier wederom voor de bandbreedte: een symmetrische, positief definiete 2×2 matrix. Het effect van deze bandbreedte is het meest duidelijk als we voor onze kernfunctie de

tweedimensionale standaard normale verdelingsfunctie kiezen: $N(0, I_2)$. $K_H\left(\begin{pmatrix}x\\y\end{pmatrix} - \begin{pmatrix}x_i\\y_i\end{pmatrix}\right)$ is dan namelijk gelijk aan de verdelingsfunctie van $N\left(\begin{pmatrix}x_i\\y_i\end{pmatrix}, H\right)$.

2.4.2 Bepalen van bandbreedte

De optimale bandbreedte hangt af van de keuze in kernfunctie en de data. De formele berekeningen hiervan zijn uitermate complex en gaan ver buiten de omvang van dit verslag. Daarom hebben we ervoor gekozen om deze af te schatten met zogeheten vuistregels en plug-in schatters.

In bijzonder hebben wij in ons onderzoek gebruik gemaakt van Scotts regel [2]. Deze methode bepaalt eerst Scotts factor c_{Scott} uit het aantal stochasten d en het aantal datapunten n. Uit deze factor en de covariantiematrix $K_{X_1,\ldots,X_d}$ wordt vervolgens een bandbreedtematrix H bepaald.

$$c_{\text{Scott}} = \frac{1}{\sqrt[d+4]{n}}$$
$$H = c_{\text{Scott}}^2 \cdot K_{X_1,\ldots,X_d}$$

Daarnaast hebben wij geëxperimenteerd met de Sheather-Jones plug-in schatter [3]. De berekening hiervan gaat buiten de omvang van dit verslag, dus zullen we niet verder toelichten. Het idee achter deze schatter is dat het aanzienlijk beter werkt dan de eerder genoemde vuistregel, maar helaas bestaat er nog geen degelijke implementatie van deze schatter in meerdere dimensies. Vandaar dat wij deze schatter niet uitvoerig hebben gebruikt in ons onderzoek.

2.4.3 Conditioneren van kernschatters

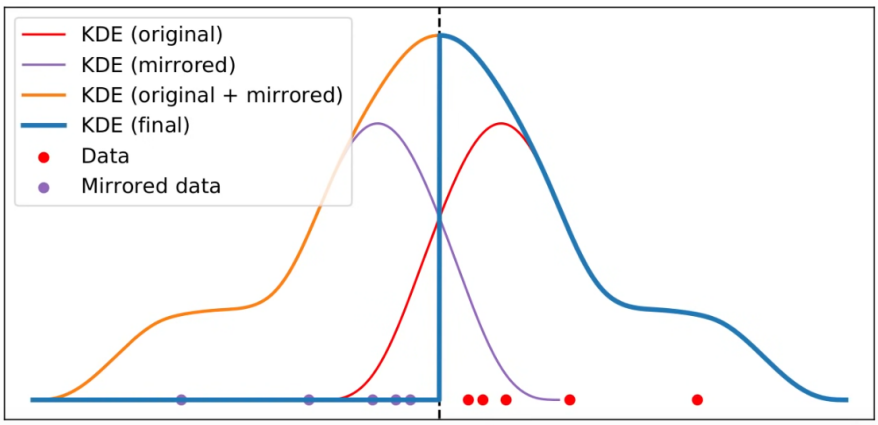
Wanneer je een kernschatter integreert over het volledige domein, dan is het resultaat per definitie 1. Een aantal keer in dit onderzoek zullen we een soort kernschatter definiëren die slechts een deel van een kansdichtheidsfunctie probeert te schatten, bijvoorbeeld alleen met domein waar $Y > c$. De originele kansdichtheidsfunctie zal op dit domein ook niet naar 1 integreren, maar naar de kans $\mathbb{P}[Y > c]$. Daarom wordt een kernschatter gebaseerd op alleen data met $Y > c$, die een deel van de gehele kansdichtheidsfunctie probeert te schatten, vermenigvuldigt met $\mathbb{P}[Y > c]$. Daarnaast is het in dit geval zinnig om het concept van spiegelen toe te passen, wat in de volgende paragraaf wordt toegelicht.

2.4.4 Spiegelen

Stel we gebruiken een kernschatter om de kansdichtheid in gebied $A = \{x \in \mathbb{R} \mid x > c\}$ te bepalen. De kernschatter kan ook een waarde groter dan nul toekennen op een punt dat niet in A ligt. Bijvoorbeeld als de kernschatter een datapunt $c+\varepsilon$ gebruikt met ε erg klein, dan is de kernfunctie positief op $c - \varepsilon$, terwijl dat buiten het gebied A ligt. Omdat we alleen integreren over waardes met $x > c$, spiegelen we de waardes met $x \leq c$ in het punt $x = c$. Als $\hat{f}_{X|X>c}$ onze kernschatter is, ziet de nieuwe gespiegelde functie er dus als volgt uit:

$$\overline{f_X}(x) = \begin{cases} \hat{f}_{X|X>c}(x) + \hat{f}_{X|X>c}(2c - x) & \text{als } x > c \\ 0 & \text{anders} \end{cases}$$

Waarbij de inputwaarde $2c-x$ volgt uit $c-|x-c| = c-(x-c) = c-x+c = 2c-x$, waarbij $|x-c|$ dus de afstand is van het punt x tot het punt waarin gespiegeld wordt. Een voorbeeld hiervan is te zien in Figuur 2.



Figuur 2: Voorbeeld Spiegelen Kernschatter [4]

Dit concept kunnen we als volgt uitbreiden naar het tweedimensionale geval. Stel we gebruiken een tweedimensionale kernschatter om de kansdichtheid in gebied $A = \{(x, y) \in \mathbb{R}^2 \mid y > c\}$ te bepalen. Wederom lopen we tegen het probleem aan dat er punten buiten A liggen met een waarde die strikt groter is dan nul. Ditmaal lossen we het probleem op door punten met $y < c$ te spiegelen in de lijn $\{(x, y) \in \mathbb{R}^2 \mid y = c\}$. Als $\hat{f}_{X,Y|Y>c}$ onze tweedimensionale kernschatter is, ziet de nieuwe gespiegelde functie er dus als volgt uit:

$$\overline{f_{X,Y}}(x, y) = \begin{cases} \hat{f}_{X,Y|Y>c}(x, y) + \hat{f}_{X,Y|Y>c}(x, 2c - y) & \text{als } y > c \\ 0 & \text{anders} \end{cases}$$

2.4.5 Nadeel kernschatters

Voordat we kernschatters toepassen voor ons probleem, zullen we eerst een nadeel toelichten. Zoals we in bovenstaande sectie zagen kan een kernschatter ook een positieve kans toekennen op waarden van y die eigenlijk een kans van 0 hebben. In onze toepassing kunnen we dit makkelijk spiegelen omdat we voor onze data weten dat $\mathbb{P}[Y \leq c] = 0$. In het algemeen weten we deze informatie niet. Dit nadeel kunnen we goed belichten door een exponentiële verdeling als voorbeeld te nemen, en deze te schatten met een kernschatter (zie Figuur 3). De rode grafiek laat een kernschatter zien met data getrokken uit de exponentiële verdeling en de blauwe de echte verdeling. We zien dat de schatting erg slecht is voor het linker uiteinde. Dit probleem is alleen niet gemakkelijk te verhelpen met spiegelen omdat we de verdeling niet aan willen nemen, dus ook niet de lijn waarin we zouden moeten spiegelen. Een kernschatter is dus niet erg goed als er opeens een kans van nul is: een discontinuïteit.

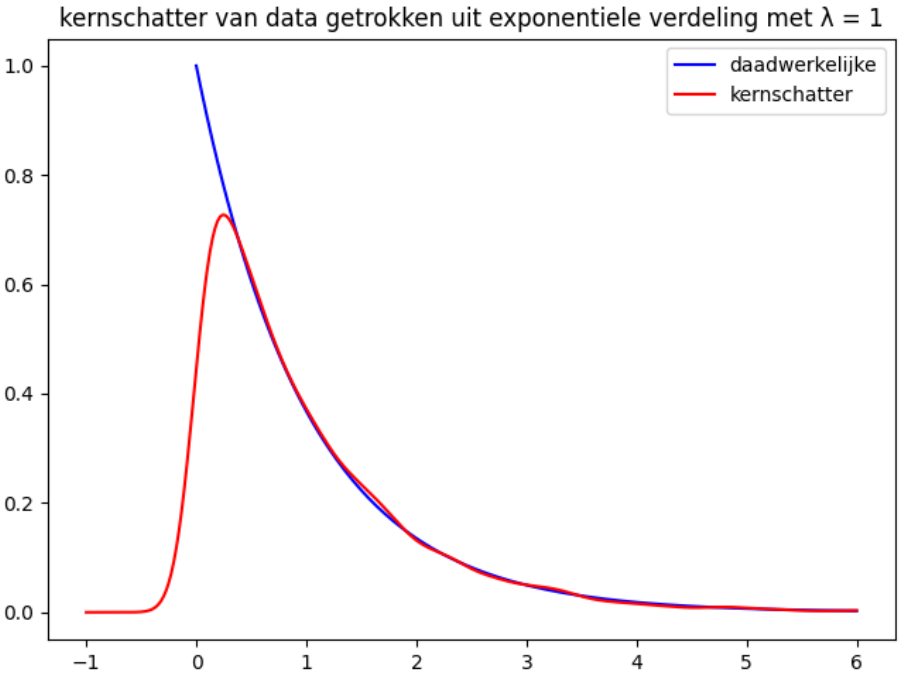
2.5 Meten van de correctheid van een schatter

Wij beschouwen drie methodes om de correctheid van een schatter te bepalen [1]:

$$\text{mean integrated squared error: } \int (f(x) - \hat{f}_X(x))^2 dx$$

$$\text{mean integrated absolute error: } \int |f(x) - \hat{f}_X(x)| dx$$

$$\text{mean integrated logarithmic error: } \int f(x) log \frac{f(x)}{\hat{f}_X(x)} dx$$

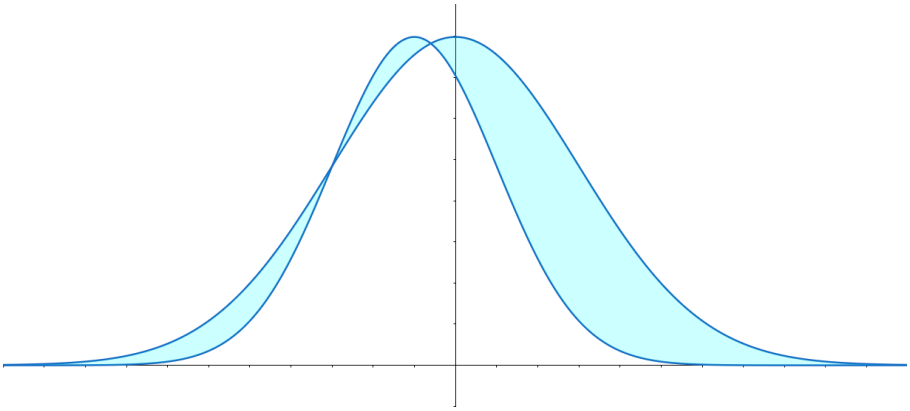


Figuur 3: Kernschatter en uiteindelijke methode toegepast op exponentiële verdeling

Voor ieder van deze integralen geldt: des te lager het resultaat, des te beter de schatter $\hat{f}_X$ is in het schatten van de daadwerkelijke verdeling f.

Hierin heeft de mean integrated squared error (MISE) een lichte voorkeur ten opzichte van de andere methodes. Het voordeel van de MISE ten opzichte van de mean integrated absolute error (MIAE), is dat de MISE grote afwijkingen van de originele verdeling zwaarder mee laat tellen in het resultaat dan de MIAE. Het voordeel van de MISE ten opzichte van de mean integrated logarithmic error (MILE), is dat de MISE is aanzienlijk makkelijker te berekenen dan de MILE. Doorgaans gebruikt men hierom vooral de MISE, en wij hebben in ons onderzoek ook uitsluitend de MISE gebruikt om de correctheid van een schatter te bepalen.

Concreet hebben wij in ons onderzoek uit iedere verdeling tachtig datasets getrokken. Uit ieder van deze datasets hebben wij een klein deel van de data gebruikt (de normale data) om een slechte schatter $\hat{f}_X(x)$ te ontwikkelen met kernschatters. Van ieder van deze slechte schatters hebben wij vervolgens de MISE berekent, waarna we het gemiddelde en de variantie van deze tachtig MISEs hebben bepaald. Dit proces hebben wij herhaald, ditmaal inclusief edge case data, voor ieder van onze methodes. De resultaten van deze slechte schatters kunnen we dan vergelijken met de resultaten van de methodes om te bepalen of onze methodes überhaupt werken, en kunnen we de resultaten van de methodes met onderling vergelijken om te bepalen welk van de methodes het beste werkt.



Figuur 4: MIAE is het blauwe oppervlak tussen de grafieken

3 Methode 1

3.1 Idee

We kunnen de data in feite splitsen in twee categorieën. Data waarvoor $Y > c$ en data waarvoor $Y \leq c$. Over de kansverdeling als $Y > c$ kunnen we meer zeggen dan over de kansverdeling als $Y \leq c$, omdat we hier meer data hebben. Daarom is het zinvol om te conditioneren op Y. Op deze manier kunnen we in het geval dat $Y > c$ veel edge case data gebruiken en in het geval dat $Y \leq c$ de normale data. The law of total probability laat ons conditioneren op Y om een kans $\mathbb{P}[X \in A]$ te bepalen. Dit geeft ons niet direct de verdeling van X, maar dit kunnen we wel krijgen door naar kansen van de vorm $\mathbb{P}[X \leq x]$ kijken, wegens de gelijkheden $F_X(x) = \mathbb{P}[X \leq x]$ en $f_X(x) = \frac{d}{dx}F_X(x)$. Hiermee kunnen we dus $F_X(x)$ en daarmee weer de gevraagde $f_X(x)$ schatten.

3.2 Formules

The law of total probability geeft ons de volgende gelijkheid:

$$\begin{aligned} F(x) &= \mathbb{P}[X \leq x] \\ &= \int_{-\infty}^{\infty} \mathbb{P}[X \leq x \mid Y = y] \cdot f_Y(y)\, dy \end{aligned}$$

en deze integraal kunnen we splitsen in het geval dat $Y > c$ en $Y \leq c$:

$$\begin{aligned} &\int_{-\infty}^{\infty} \mathbb{P}[X \leq x \mid Y = y] \cdot f_Y(y)\, dy \\ &\quad = \int_{-\infty}^{c} \mathbb{P}[X \leq x \mid Y = y] \cdot f_Y(y)\, dy + \int_{c}^{\infty} \mathbb{P}[X \leq x \mid Y = y] \cdot f_Y(y)\, dy \end{aligned}$$

Door deze integralen te schatten kunnen we dus een schatter $\hat{F}_X$ bepalen. Vervolgens kunnen we door middel van numerieke differentiatie hiermee $\hat{f}_X$ bepalen. We lichten stap voor stap toe hoe we de integralen schatten.

3.2.1 De eerste integraal

Eerst kan de integrand van de eerste integraal vereenvoudigd worden:

$$\begin{aligned} \mathbb{P}[X \leq x \mid Y = y] \cdot f_Y(y) &= \frac{\int_{-\infty}^{x} f_{X,Y}(u,y)du}{\int_{-\infty}^{\infty} f_{X,Y}(u,y)du} \cdot \int_{-\infty}^{\infty} f_{X,Y}(u,y)du \\ &= \int_{-\infty}^{x} f_{X,Y}(u,y)du \end{aligned}$$

Het product is dus gelijk aan $\int_{-\infty}^{x} f_{X,Y}(u,y)du$. Aangezien we hier alleen kijken naar $f_{X,Y}(x,y)$ met $y \leq c$, kunnen we hier het beste $f_{X,Y}$ schatten met behulp van de normale data. De normale data is verdeeld met kansverdeling $f_{X,Y}$ dus dit kan eenvoudig door middel van een kernschatter. De details hiervan worden besproken bij de implementatie. Merk op dat deze schatter niet op erg veel data gebaseerd is en dus vooral in de extremen niet zo nauwkeurig meer is. Daarom willen we deze schatter ook alleen gebruiken als $y \leq c$.

3.2.2 De tweede integraal

Voor de tweede integraal worden $\mathbb{P}[X \leq x \mid Y = y]$ en $f_Y(y)$ wel apart geschat. Eerst wordt toegelicht hoe we $\mathbb{P}[X \leq x \mid Y = y]$ schatten.

Als we heel veel data zouden hebben met precies $Y = y$ kunnen we $\mathbb{P}[X \leq x \mid Y = y]$ makkelijk schatten door $\frac{\#\text{punten met } X \leq x \text{ en } Y = y}{\#\text{punten met } Y = y}$. Echter zullen we over het algemeen nooit veel data

hebben met precies $Y = y$, omdat Y een continue stochast is. Daarom tellen we in plaats daarvan de punten waarbij Y dicht bij y ligt door middel van de formule $\frac{\#\text{punten met } X \leq x \text{ en } |Y-y|<\varepsilon}{\#\text{punten met } |Y-y|<\varepsilon}$. Daarvoor moeten we een bandbreedte ε bepalen die bepaalt wat als dichtbij telt. We willen de bandbreedte zo kiezen dat er gemiddeld n punten liggen in het interval waarin we de punten tellen. Wij hebben voor $n = 100$ gekozen, dit is namelijk een redelijk aantal punten om de kans te kunnen schatten, maar ook klein genoeg zodat ε niet te groot wordt. Dit kan door de lengte van het interval gelijk te stellen aan het totale interval waarin punten met $Y > c$ voorkomen gedeeld door het aantal punten dat in dit totale interval voorkomen vermenigvuldigd met 100. Dit geeft de volgende formule:

$$2\varepsilon = 100 \cdot \frac{Y_{(N+K)} - c}{\#\text{punten met } (Y > c)}$$

De teller is de afstand tussen c en de grootste waarde van Y die voorkomt in de data. De noemer is het aantal punten met $Y > c$, oftewel het aantal punten in $(c, Y_{(N+K)}]$. We noemen de functie waarmee we $\mathbb{P}[X \leq x \mid Y = y]$ schatten g. Hiervoor geldt dus:

$$g(x, y) \coloneqq \frac{\#\text{punten met } X \leq x \text{ en } Y \in (y-\varepsilon, y+\varepsilon)}{\#\text{ punten met } Y \in (y-\varepsilon, y+\varepsilon)}$$

Nu moeten we alleen $f_Y(y)$ nog schatten voor $y > c$. Ook dit willen we gaan doen door middel van een kernschatter, maar dan die alleen gebaseerd is op de edge case data. We kunnen namelijk ook een kernschatter maken die slechts een deel van de gezochte kansdichtheid schat. In dit geval willen we f_Y alleen schatten op het interval $[c, \infty)$, (zie 2.4.4). Daarvoor hebben we de kans $\mathbb{P}[Y > c]$ nodig (zie 2.4.3). Omdat er N datapunten voldoen aan $Y > c$ en $N + M$ datapunten in totaal waren getrokken, kunnen we dit schatten met $\frac{N}{N+M}$. Dit geeft dus een kernschatter $\hat{f}_Y$, die alleen gedefinieerd is op het interval $[c, \infty)$.

3.2.3 De schatter voor $\hat{f}_X$

Deze dingen samen genomen geeft de volgende formule waarmee we $f_X(x)$ benaderen:

$$\hat{f}_X(x) \coloneqq \frac{d}{dx}\left(\int_{-\infty}^{c}\int_{-\infty}^{x} \hat{f}_{X,Y}(u, y)\, du\, dy + \int_{c}^{\infty} g(x, y) \cdot \hat{f}_Y(y) dy\right) \tag{1}$$

Hierbij moet de $\frac{d}{dx}$ geïnterpreteerd worden als numerieke differentiatie, omdat het niet te doen zal zijn dit analytisch te differentiëren. Evenzo worden alle integralen numeriek bepaald. Hiervan staan de details in de implementatie.

3.3 Implementatie

In deze paragraaf wordt de implementatie toegelicht. Als eerst wordt er 2D data gegenereerd. In het script in de appendix wordt er normaal verdeelde data gegenereerd, maar andere verdelingen zijn ook mogelijk. De edge case data, normale data, N en M worden gedefinieerd zoals beschreven in ons probleem. Van de originele data wordt $\frac{N}{10}$ datapunten geselecteerd als normale data, zodat er in verhouding altijd meer edge case data is.

De schatter $\hat{f}_{X,Y}$ wordt gemaakt door met `SciPy` een 2D kernschatter te genereren met de functie `gaussian_kde`. Deze maakt gebruik van Scott's factor voor de bandbreedte. De schatter noemen we `bad_estimator`. Hiermee kan de integraal van $-\infty$ to c in (1) berekend worden. Voor de integraal van c to ∞ wordt eerst de functie g gedefinieerd door de Pythonfunctie `probability_X_le_x_given_y`. Voor $\hat{f}_Y(y)$ wordt eerst een normale kernschatter gemaakt op basis van de Y-coördinaten in de edge case data. De bandbreedte van deze kernschatter wordt bepaald door de de Sheather-Jones schatter. Deze kernschatter wordt pas gespiegeld in de lijn $y = c$ en vermenigvuldigt met $\frac{N}{N+M}$ zoals beschreven in de methode op de plek waar de integraal uitgerekend wordt.

In de functie `F` wordt $\hat{F}(x)$ berekend. De integralen kunnen gedaan worden met de `SciPy` functies `dblquad` voor een tweedimensionale integraal en `quad` voor een normale integraal. Deze `F` moet nu gedifferentieerd worden om $\hat{f}(x)$ te krijgen. Dit wordt numeriek gedaan met de methode `np.gradient`. Deze methode berekent $\frac{\hat{F}(x_0)-\hat{F}(x_1)}{x_0-x_1}$ voor x_0, x_1 die dicht bij elkaar liggen. Dit zal dus de afgeleide benaderen en daarmee wordt $\hat{f}(x)$ verkregen. De volledige implementatie is te vinden in appendix A.

4 Methode 2

4.1 Idee

Analoog aan de vorige methode, schatten we bij deze methode weer kansen van de vorm $\mathbb{P}[X \leq x]$ om F_X te kunnen schatten. In plaats van de total law of probability te gebruiken, berekenen we deze kans nu door middel de van de tweedimensionale kansdichtheidsfunctie $f_{X,Y}$, zoals beschreven in 2.1. Bij deze methode gaan we dus de tweedimensionale kansdichtheidsfunctie schatten. Weer kunnen we de integraal opsplitsen in een deel waarbij $y \leq c$ en $y > c$. In het geval $y \leq c$ schatten we deze kansdichtheid met de normale data en in het geval $y > c$ met de edge case data.

4.2 Formules

We berekenen $F_X(x)$ door middel van de tweedimensionale kansdichtheidsfunctie $f_{X,Y}$ als volgt:

$$\begin{aligned}
F_X(x) &= \mathbb{P}[X \leq x] \\
&= \mathbb{P}[(X,Y) \in (-\infty, x] \times \mathbb{R}] \\
&= \int_D f_{X,Y}(u,y)dudy \qquad \text{(waarin } D = (-\infty, x] \times \mathbb{R}) \\
&= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{x} f_{X,Y}(u,y)du \right) dy \\
&= \int_{-\infty}^{c} \left(\int_{-\infty}^{x} f_{X,Y}(u,y)du \right) dy + \int_{c}^{\infty} \left(\int_{-\infty}^{x} f_{X,Y}(u,y)du \right) dy
\end{aligned}$$

De eerste integraal is hetzelfde als de eerste integraal in methode 1 nadat de integrand versimpeld was. Deze wordt dus ook op dezelfde manier geschat. Ter herinnering, we maken met de normale data een kernschatter $\hat{f}_{X,Y}$. Deze schatter is vooral in extreme gevallen niet erg nauwkeurig, omdat het gebaseerd is op weinig data. Vervolgens integreren we deze schatter numeriek om de eerste term te bepalen.

Voor de tweede integraal willen we ook $f_{X,Y}(x,y)$ schatten, maar nu alleen voor $y > c$. Dit kunnen we net als bij methode 1 doen door middel van een kernschatter die gebaseerd is op alleen de edge case data. Dit komt dus op hetzelfde neer als bij methode 1, maar dan met een tweedimensionale kernschatter, zoals ook beschreven in 2.1. Hiervoor hebben we ook weer de kans $\mathbb{P}[Y > c]$ nodig, die we nog steeds met $\frac{N}{N+M}$ schatten. Dit geeft dus een kernschatter $\hat{f}_{X,Y|Y>c}$, die alleen gedefinieerd is op het domein $\mathbb{R} \times [c, \infty)$. De tweede integraal schatten we dus door $\hat{f}_{X,Y|Y>c}$ numeriek te integreren.

Al met al krijgen we de volgende schatter voor f_X:

$$\hat{f}_X \coloneqq \frac{d}{dx} \left(\int_{-\infty}^{c} \left(\int_{-\infty}^{x} \hat{f}_{X,Y}(u,y)du \right) dy + \int_{c}^{\infty} \left(\int_{-\infty}^{x} \hat{f}_{X,Y|Y>c}(u,y)du \right) dy \right) \tag{2}$$

Ook hier moeten de differentiatie en integratie geïnterpreteerd worden alsof deze numeriek worden gedaan. Dit wordt verder toegelicht in de volgende paragraaf over de implementatie.

4.3 Implementatie

In deze paragraaf wordt de implementatie toegelicht. De implementatie lijkt in veel opzichten op die van methode 1. De data generatie en initialisatie van de edge case data, normale data, N en M gaat precies hetzelfde als bij methode 1. Ook de schatter $\hat{f}_{X,Y}$ wordt op dezelfde manier bepaald. Hiermee kan de integraal van $-\infty$ to c in (2) berekend worden.

De schatter $\hat{f}_{X,Y|Y>c}$ wordt ook met `SciPy` gemaakt, maar nu op basis van de edge case data. Deze

kernschatter zal echter tot 1 integreren, daarom wordt deze vermenigvuldigd met $\frac{N}{N+M} \approx \mathbb{P}[Y > c]$. Ook wordt het spiegelen toegepast. Beide toepassingen worden pas gedaan bij het berekenen van de integraal.

De twee integralen die berekend zijn met `dblquad` worden bij elkaar opgeteld om $\hat{F}(x)$ te verkrijgen. Deze moet nu gedifferentieerd worden om $\hat{f}(x)$ te krijgen. Dit wordt numeriek gedaan met `np.gradient`, analoog aan methode 1. De volledige implementatie is te vinden in appendix B.

5 Methode 3

5.1 Idee

Normaal bij kernschatters vermenigvuldigt men elke kernfunctie met $\frac{1}{N}$, waarbij N het aantal datapunten is. Dit betekent in feite dat elk punt even zwaar meeweegt. Het idee is nu om te compenseren voor de scheef verdeelde data door in bepaalde punten met een lagere constante te vermenigvuldigen dan in andere punten. De edge case data is als het ware overgerepresenteerd in de data. Dit kan dus precies weer gecompenseerd worden door de kernfunctie van deze data te vermenigvuldigen met een lagere constante. Uiteraard moet het geheel nog steeds integreren tot 1.

5.2 Formules

We beschouwen eerst het normale geval van een kernschatter. Stel dat we data $\{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ hebben dat niet scheef verdeeld is. Dan kunnen we de verdeling van X met een kernschatter als volgt benaderen:

$$\hat{f}_X(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - X_i)$$

Deze som zouden we op kunnen splitsen in data waarbij $Y > c$ en $Y \leq c$:

$$\begin{aligned}\hat{f}_X(x) &= \frac{1}{n}\sum_{i=1}^{n} K_h(x - X_i) \\ &= \frac{1}{n}\sum_{\substack{1\leq i\leq n \\ Y_i \leq c}} K_h(x - X_i) + \frac{1}{n}\sum_{\substack{1\leq i\leq n \\ Y_i > c}} K_h(x - X_i)\end{aligned}$$

We zien dat de term met de eerste sommatie integreert naar $\frac{\#\text{punten met} Y_i \leq c}{\text{totaal aantal punten}} \approx \mathbb{P}[Y_i \leq c]$, omdat elke kernfunctie integreert naar 1. Evenzo integreert de tweede sommatie naar ongeveer $\mathbb{P}[Y_i \leq c]$. Als we nu scheef verdeelde data hebben, maar de constante $\frac{1}{n}$ voor beide sommaties zo veranderen dat de sommaties nog steeds naar deze kansen integreren, compenseren we precies voor de scheef verdeelde data.

Dit gaan we nu dus toepassen op de normale data en de edge case data. Omdat deze deels overlappen en het voor deze methode ongewenst is om punten dubbel te tellen filteren we uit de normale data de punten met $Y > c$, deze komen namelijk al voor in de edge case data. We schrijven de overgebleven normale data als $\left\{\begin{pmatrix} X_i \\ Y_i \end{pmatrix}\right\}_{i\in Norm}$ en de edge case data als $\left\{\begin{pmatrix} X_i \\ Y_i \end{pmatrix}\right\}_{i\in Edge}$. De hoeveelheid overgebleven normale data $|Norm|$ noemen we K'. De aangepaste kernschatter voor deze data wordt dan:

$$\hat{f}_X(x) = c_1 \sum_{i\in Norm} K_h(x - X_i) + c_2 \sum_{i\in Edge} K_h(x - X_i)$$

Nu willen we constanten c_1 en c_2 zo bepalen dat $c_1 \sum_{i\in Norm} K_h(x - X_i)$ sommeert naar $\mathbb{P}[Y_i \leq c]$ en $c_2 \sum_{i\in Edge} K_h(x - X_i)$ naar $\mathbb{P}[Y_i > c]$.

De kansen $\mathbb{P}[Y_i \leq c]$ en $\mathbb{P}[Y_i > c]$ zijn niet gegeven dus ook deze moeten we schatten. $\mathbb{P}[Y_i > c]$ schatten we zoals eerder met $\frac{N}{N+M}$ en $\mathbb{P}[Y_i \leq c]$ schatten we dus met $\frac{M}{N+M}$. $c_1 \sum_{i\in Norm} K_h(x - X_i)$ integreert naar $c_1 \cdot K'$. We kiezen dus $c_1 = \frac{M}{(N+M)\cdot K'}$, zodat $c_1 \sum_{i\in Norm} K_h(x - X_i)$ integreert naar $\frac{M}{N+M}$, onze schatter voor $\mathbb{P}[Y_i \leq c]$. c_2 kiezen we op dezelfde manier dus $c_2 = \frac{N}{(N+M)\cdot N} = \frac{1}{N+M}$, zodat $c_2 \sum_{i\in Edge} K_h(x - X_i)$ integreert naar $\frac{1}{N+M} \cdot N = \frac{N}{N+M}$, onze schatter voor $\mathbb{P}[Y_i > c]$. Verder hebben we mogelijk voor de eerste sommatie een andere hoeveelheid data dan voor de tweede

sommatie. Het kan daarom zijn dat er verschillende bandbreedtes voor de kernfuncties het beste werken. Hoe we deze bandbreedtes kiezen, wordt toegelicht in de uitleg over de implementatie. We krijgen de volgende schatter voor f_X:

$$\hat{f}_X(x) = \frac{M}{(N+M)\cdot K'} \sum_{i \in Norm} K_h(x - X_i) + \frac{1}{N+M} \sum_{i \in Edge} K_{h'}(x - X_i) \tag{3}$$

5.3 Implementatie

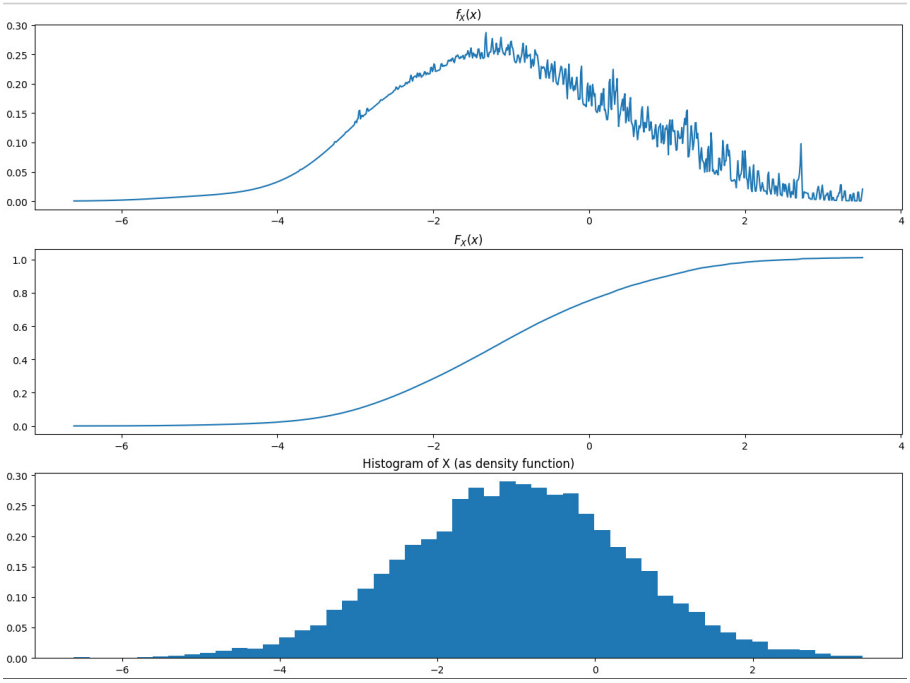
In deze paragraaf wordt de implementatie toegelicht. Het genereren van data en de initialisatie van de edge case data, normale data, N en M gaat precies hetzelfde als bij methodes 1 en 2. Vervolgens nemen we een nieuwe lijst waarin we alleen de X-coördinaten nemen van de normale data waarbij $Y \leq c$. Dit is dan de gefilterde normale data. Ook van de edge case data maken we een nieuwe lijst waarin we alleen de X-coördinaten nemen, omdat we de Y-coördinaten niet meer gebruiken.

We maken twee kernschatters met de `gaussian_kde` functie, één op basis van de gefilterde normale data (`left_kde`) en één op basis van de edge case data (`right_kde`). De bandbreedtes worden bepaald door Scotts regel. Ze zijn dus verschillend, omdat de kernschatters beide zijn gebaseerd op andere data. De kernschatter gebaseerd op de gefilterde normale data is nu gelijk aan $\frac{1}{K'} \sum_{i \in Norm} K_h(x - X_i)$ en de kernschatter gebaseerd op de edge case data is gelijk aan $\frac{1}{N} \sum_{i \in Edge} K_h(x - X_i)$. Om de schatter in (3) te krijgen, moeten we dus $\frac{M}{N+M} \cdot$ `left_kde` $+ \frac{N}{N+M} \cdot$ `right_kde` nemen. De volledige implementatie is te vinden in appendix C.

6 Analyse en resultaten

Alle methodes zijn vergeleken met kernschatters die gebruik maakten van $\frac{N}{10}$ datapunten getrokken uit een normale verdeling en met de werkelijke kansdichtheid. Op deze manier kon er dus gekeken worden in hoeverre een 'slechte' schatter verbeterd zou worden en of deze op de werkelijke kansdichtheid lijkt. De data is hier altijd getrokken uit normale verdelingen waarbij $\mu_x = 0,\ \mu_y = 0,\ \sigma_x = 1,\ \sigma_y = 1$ en $cov_{x,y} = 0.8$

6.1 Methode 1



Figuur 5: Resultaten methode 1

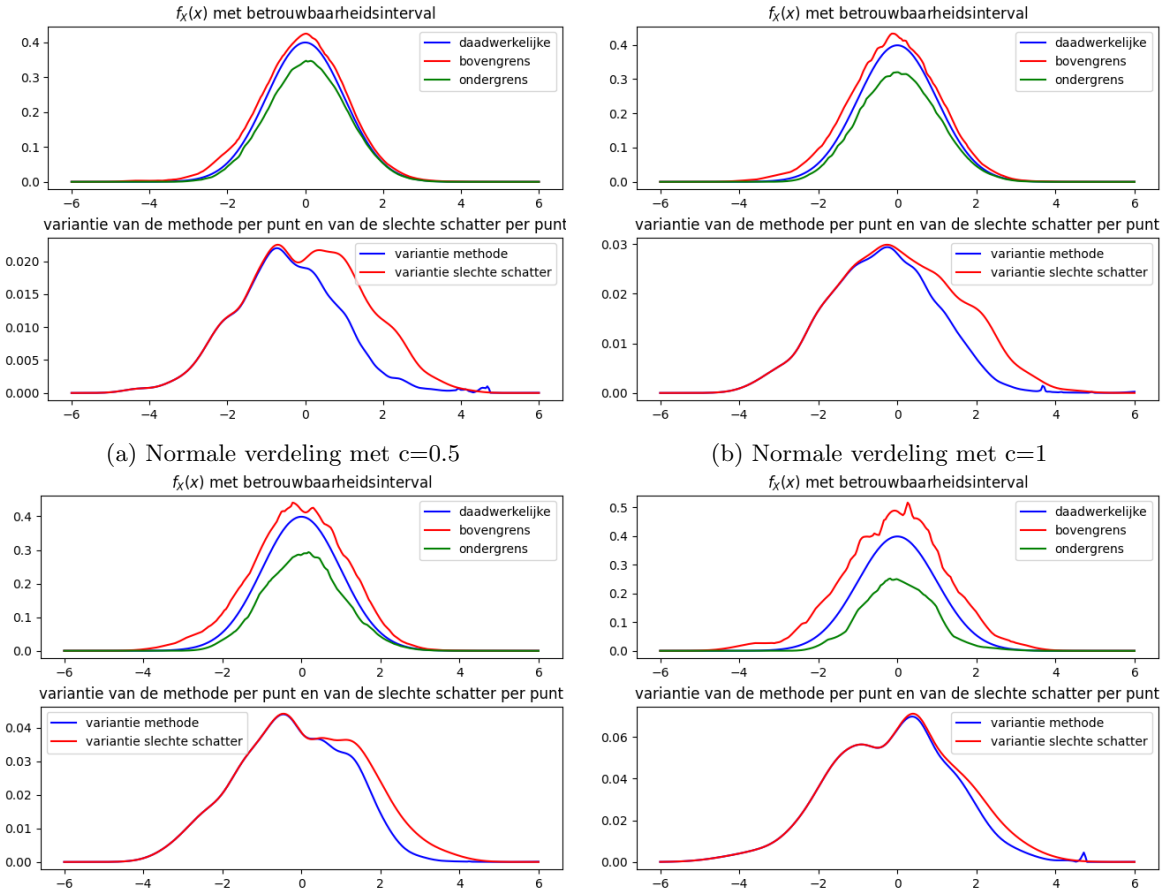
In bovenstaand figuur is de schatter van methode 1 weergegeven met het bijbehorende histogram van de data. Zoals te zien is, is $\hat{f}_X(x)$ niet glad. Waarschijnlijk is dit te wijten aan de functie g die niet continu is. Omdat deze schatter meerdere keren niet glad werd en wij de oorzaak wel dachten te duiden, zijn wij niet verder gegaan met het onderzoeken van deze methode. Het maken van een schatter met deze methode duurde ongeveer vijf minuten met onze implementatie.

6.2 Methode 2

Figuur 6 bestaat uit twee grafieken per verschillende waarden van c. De schatters zijn per waarde van x gesorteerd, en vervolgens zijn de bovenste en onderste 2,5 procent verwijdert. De bovengrens is de hoogste overgebleven waarde hiervan en de ondergrens, de kleinste waarde. Deze boven- en ondergrens geven dus een 95 procent betrouwbaarheidsinterval voor onze schatter. De onderste grafieken uit Figuur 6 geven de variantie tussen de 80 verschillende schatter die gegenereerd zijn en de variantie tussen de 80 verbeterde schatters (de uitkomst van methode 2).

Het is met name goed te zien dat de schatter vooral aan de rechterkant verbeterd wordt, aangezien het betrouwbaarheidsinterval en de variantie rechts kleiner zijn. Dat is ook te verwachten aangezien X en Y positief gecorreleerd zijn en we meer data hebben met $Y > c$. Deze schatters liggen dichter bij elkaar en ook dichter bij de daadwerkelijke waarden. Ook is goed te zien dat

de verbetering minder wordt naarmate c groter wordt. Al blijft de daadwerkelijke waarde mooi binnen de boven- en ondergrens blijft, zien we dat de variantie erg toeneemt aan de rechterkant. Dit wijten wij aan het feit dat er weinig edge case data is omdat de kans $\mathbb{P}[Y > c]$ kleiner wordt voor grotere c. Het maken van een schatter met deze methode duurde ongeveer een uur met onze implementatie.



(a) Normale verdeling met c=0.5

(b) Normale verdeling met c=1

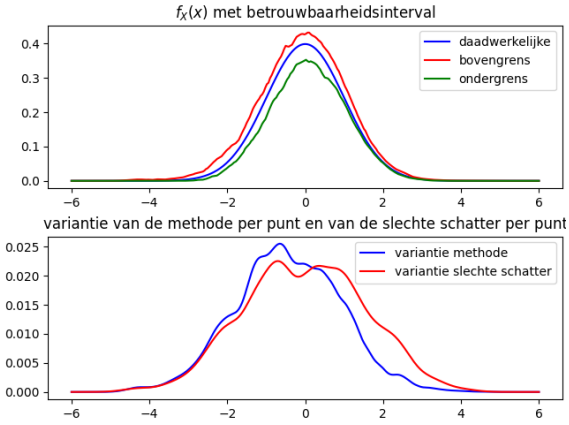
(c) Normale verdeling met c=1.5

(d) Normale verdeling met c=2

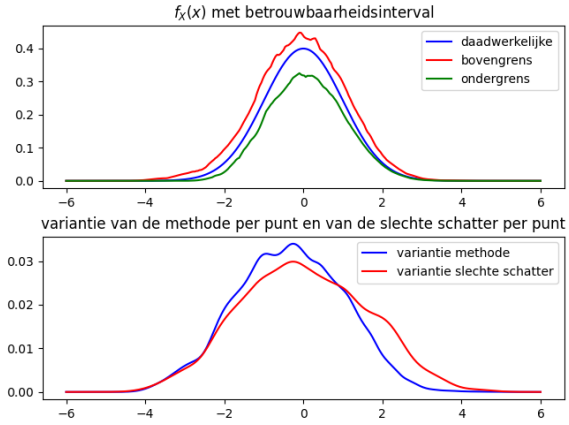
Figuur 6: Resultaten Methode 2

6.3 Methode 3

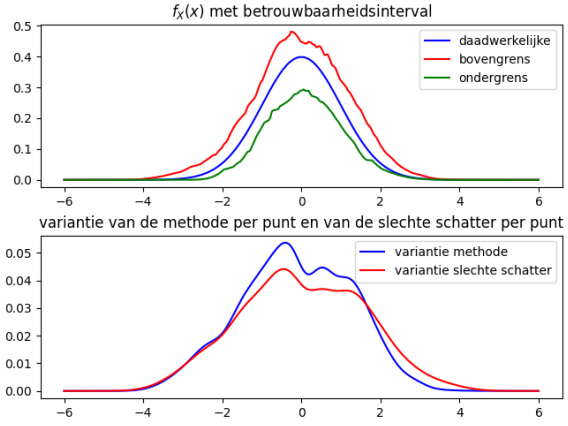
In Figuur 7 zijn de resultaten weergegeven op dezelfde manier als beschreven in sectie 6.2. Wederom is te zien dat de schatters aan rechterkant verbeterd worden, al is dit in mindere mate dan bij methode 2. De ondergrens lijkt in meerdere grafieken erg dichtbij de werkelijke waarde te liggen, terwijl de bovengrens er wat meer vanaf ligt. Ook valt op dat de variantie van deze schatter groter is aan de linkerkant en rond 0 dan de variantie van de slechte schatter. We vonden het moeilijk dit verschijnsel te verklaren. Het vermoeden is dat het met het bepalen van de bandbreedte van de kernschatters te maken heeft, maar ook dat zou het niet helemaal verklaren. Het maken van een schatter met deze methode duurde ongeveer twee seconden met onze implementatie.



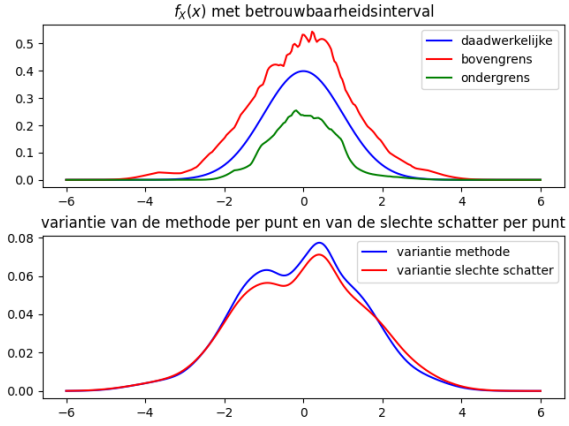
(a) Normale verdeling met c=0.5



(b) Normale verdeling met c=1



(c) Normale verdeling met c=1.5



(d) Normale verdeling met c=2

Figuur 7: Resultaten Methode 3

c	slechte schatter	methode 2	methode 3
0.5	0.00019440	0.00013986	0.00015182
1.0	0.00034020	0.00029385	0.00029837
1.5	0.00066260	0.00058082	0.00068909
2.0	0.00145251	0.00136398	0.00152118

Figuur 8: Mean integrated squared errors

6.4 Kwantificatie methodes

Om de correctheid van onze methodes te kwantificeren hebben we de Mean Integrated Square Error van de geschatte kansdichtheidfuncties bepaald van de 80 eerder genoemde grafieken. Dit hebben we ook gedaan voor 80 kernschatters gebaseerd op slechts de normale data (zie 2.5). De gemiddelde resultaten zijn gegeven in bovenstaande tabel. Zoals te zien is, geeft methode 2 een lagere waarde dan de slechte schatter, wat betekent dat het de echte kansdichtheid beter schat. Methode 3 geeft lagere waarden dan de slechte schatter voor c=0.5 en c=1.0, en hogere waarden voor 1.5 en 2.0. We hebben eerder al benoemd dat de variantie bij deze methode voor grotere c minder klein wordt dan de andere methode en bij kleinere waarden van c. Bij grotere waarden van c is de dataset kleiner, en daardoor wordt de bandbreedte slechter bepaald. Dit bevestigt ons vermoeden dat het bepalen van de bandbreedte bij deze methodes problemen oplevert.

6.5 Generaliseren van de methodes

In deze paragraaf zullen we kort nagaan hoe methodes 2 en 3 in een algemener geval ingezet kunnen worden, waarin de data bestaat uit meer dan twee componenten. Methode 1 hebben we achterwege gelaten, omdat deze geen goede resultaten opleverde.

Methode 2 kan ook werken in hogerdimensionale gevallen en edge case criteria van (bijna) willekeurige vorm. Laten X_1, X_2, ..., X_n de reëelwaardige stochasten zijn waarvan we geïnteresseerd zijn in de verdeling. Laat Y de d-dimensionale stochast zijn die bepaalt wanneer data onder edge case data valt. Merk op dat als we geïnteresseerd zijn in de verdeling van Y we prima kunnen nemen dat één of meerdere X_i's gelijk is of zijn aan componenten van Y. Laten we zeggen dat een datapunt $(X_1, \ldots, X_n, Y)$ voldoet aan het edge case criterium als $Y \in E$, voor één of andere verzameling $E \subset \mathbb{R}^d$. De simultane kansverdeling van X_1, ..., X_n voldoet aan:

$$\begin{aligned}
F_{X_1,\ldots,X_n}(x_1,\ldots,x_n) &= \mathbb{P}[X_1 \leq x_1, \ldots, X_n \leq x_n] \\
&= \mathbb{P}[X_1 \leq x_1, \ldots, X_n \leq x_n, Y \in \mathbb{R}^d] \\
&= \int_D f_{X_1,\ldots,X_2,Y}(u_1,\ldots,u_2,y)\, du_1 \ldots du_n dy \\
&= \int_E \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_n} f_{X_1,\ldots,X_2,Y}(u_1,\ldots,u_2,y)\, du_1 \ldots du_n dy \\
&\quad + \int_{E^c} \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_n} f_{X_1,\ldots,X_2,Y}(u_1,\ldots,u_2,y)\, du_1 \ldots du_n dy
\end{aligned}$$

Hierin geldt $D = \{(u_1, \ldots, u_n, y) \in \mathbb{R}^{d+n} |\ \forall_i\ u_i \leq x_i\}$. $f_{X_1,\ldots,X_2,Y}$ kan in de eerste integraal nu geschat worden door middel van de edge case data met behulp van een hogerdimensionale kernschatter. Deze vermenigvuldigen we met een schatting van $\mathbb{P}[Y \in E]$, die we nog steeds kunnen maken met $\frac{N}{N+M}$ omdat we nog steeds weten hoeveel edge case data we hebben en hoeveel data er is weggegooid. Afhankelijk van wat voor vorm E heeft, is het echter niet altijd meer mogelijk om het concept van spiegelen toe te passen. Als Y tweedimensionaal is en het edge criterium is om aan één kant van een lijn te liggen is het bijvoorbeeld nog wel mogelijk, maar zodra E wat gekkere vormen krijgt, wordt het erg lastig of onmogelijk.

In de tweede integraal kan net als in de originele methode $f_{X_1,\ldots,X_2,Y}$ geschat worden door middel van een kernschatter gebaseerd op de normale data. Het integreren kan gemakkelijk numeriek gedaan worden. Tot slot geldt er $f_{X_1,\ldots,X_n}(x_1,\ldots,x_n) = \frac{\partial^n F_{X_1,\ldots,X_n}(x_1,\ldots,x_n)}{\partial x_1 \ldots \partial x_n}$. Dus met de schatting van $F_{X_1,\ldots,X_n}$ kunnen we door numerieke partiële differentiatie de kansdichtheidsfunctie $f_{X_1,\ldots,X_n}$ schatten.

In vele algemenere gevallen kan methode 2 dus ook toegepast worden. Echter als het edge case criterium van een ingewikkelde vorm is, kan het spiegelen niet meer toegepast worden, waardoor

er een beetje nauwkeurigheid verloren gaat. Verder moet ook opgemerkt worden dat naarmate er meer dimensies zijn, de implementatie van deze methode vele malen trager zal worden, omdat het uitrekenen van hogerdimensionale integralen steeds langzamer gaat. Methode 3 is wat dat betreft mogelijk een betere optie voor gevallen met hogere dimensies. We zullen nu bespreken hoe methode 3 in dit geval ingezet kan worden.

Ook methode 3 kan ingezet worden in het algemene geval. We moeten nu wel n-dimensionale kernfuncties nemen. Er verandert eigenlijk heel weinig aan de methode. $Norm$ is nu gedefinieerd als $\{i \in \{1, \ldots, K\} |\ Y_i \notin E\}$ (ter herinnering, K was de hoeveelheid normale data) en nog steeds geldt $K' = |Norm|$. We krijgen de volgende schatter:

$$\begin{aligned}\hat{f}_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = &\frac{M}{(N+M) \cdot K'} \sum_{i \in Norm} K_h((x_1 - X_1, \ldots, x_n - X_n)) \\ &+ \frac{1}{N+M} \sum_{i \in Edge} K_{h'}((x_1 - X_1, \ldots, x_n - X_n))\end{aligned}$$

We zijn in principe niet eens gelimiteerd tot "normale data"en "edge case data". We kunnen ook nog meer categorieën maken mits we de kansen weten of kunnen schatten dat Y in een specifieke categorie zit. Laten we de waarden die Y kan hebben opdelen in E_1,...,E_k voor één of andere k, dus zodat E_1,...,E_k disjunct zijn en $E_1 \cup \cdots \cup E_n = \mathbb{R}^d$. We trekken data uit de verdelingen met kansdichtheden $f_{X_1,\ldots,X_n,Y|Y \in E_i}$. Voor elke i kan dit een andere hoeveelheid data zijn, de data mag dus scheef verdeeld zijn.

$$\hat{f}_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = \sum_{j=1}^{k} \left(\frac{\mathbb{P}[Y_i \in E_1]}{|\{i | Y_i \in E_1\}|} \sum_{i, Y_i \in E_1} K_{h_j}((x_1 - X_1, \ldots, x_n - X_n)) \right)$$

De formule komt neer op een aparte kernschatter per categorie, vermenigvuldigd met een constante zodat het integreert naar $\mathbb{P}[Y_i \in E_1]$. Eventueel kan voor elke kernschatter weer een andere bandbreedte gekozen worden.

7 Conclusie

Gezocht werd naar een methode om een dataset met edge case data op te nemen in een normale dataset, om zo de kansdichtheid van je data beter te kunnen schatten. Om dit te bereiken hebben we drie methodes bedacht en uitgewerkt in het geval de data uit een tweedimensionale normale verdeling komt.

Methode 1, die gebaseerd is op de law of total probability, bleek hier niet toe in staat. Al kwam deze schatter in de buurt van de werkelijke kansverdeling, was de kansdichtheid die door deze methode voortgebracht werd niet glad.

Methode 2 is ook gebaseerd op de law of total probability, maar maakt beter gebruik van kernschatters. Deze methode is een goede kandidaat voor het verbeteren van een schatter voor de kansdichtheid. Zo hebben we aangetoond dat deze methode zowel het betrouwbaarheidsinterval, als de variantie en de mean integrated squared error verkleint ten opzichte van een kernschatter die gebaseerd is op alleen de normale data. Echter moet er een kanttekening geplaats worden dat het berekenen van de schatter met deze methode veel tijd kost. Als dit een beperking is kan mogelijk beter naar andere methodes gekeken worden.

Dit brengt ons tot Methode 3. Deze methode splitst de dataset op in de normale data en edge case data, neemt een kernschatter over beide delen en neemt daarover een genormeerde som. Zoals te zien in de grafieken was de verbetering met name te zien in de staart waar de edge case data zich bevond. Deze methode zouden we aanraden als de schatter voornamelijk verbeterd moet worden op het gebied waar de edge case data zich bevindt en als het tijdsbestek van korte duur is, aangezien het berekenen van deze schatter veel minder tijd in beslag neemt.

Methodes 2 en 3 zijn ook uitgewerkt voor de gevallen dat de data meer dan 2 dimensionaal is, en het geval dat de edge case data een andere vorm betreft. Methode 3 is veel eenvoudiger te veralgemeniseren, al zou dit voor methode 2 ook moeten werken. Bij methode 2 zouden we in hogere dimensies het spiegelen mogelijk niet kunnen toepassen, we weten niet hoeveel slechter dit de schatter zou maken aangezien we deze methode niet hebben geïmplementeerd. Ook voor methode 3 hebben we voor de veralgemenisering geen implementatie gemaakt.

Over het algemeen hebben we dus een goede verbetering gezien door de methodes op het gebied waar de edge case data zich bevind. Verder zouden we Methode 2 aanbevelen omdat deze methode de meeste verbetering liet zien. Ook belangrijk om te benoemen is dat de duur van schatters bepalen met Methode 2, niet sterk afhankelijk is van het aantal datapunten, omdat de integraal de bepalende factor is en niet het bepalen van de kernschatter zelf.

8 Discussie

Over het algemeen zijn wij erg tevreden over het gehele proces van ons onderzoek, toch zijn er een aantal punten die we willen belichten.

8.1 Verdelingen

We hebben om onze methodes te testen gebruik gemaakt van data uit twee verdelingen. De exponentiële verdeling hebben we niet in dit verslag laten zien omdat dit niet erg uitgebreid was en geen nieuwe inzichten gaf. Dit hadden er misschien meer kunnen zijn, om een uitgebreider onderzoek te hebben. Denk hier bijvoorbeeld aan een verdeling met meerdere pieken, X en Y uit verschillende verdelingen of meer discontinue verdelingen. Dit laatste om het eerder genoemde mankement van kernschatters te belichten.

8.2 Bandbreedte

Het resultaat van een kernschatter is sterk afhankelijk van de gekozen bandbreedte. Echter viel de keuze van de bandbreedte buiten de scope van ons onderzoek. In de `SciPy` library wordt standaard gebruik gemaakt van Scotts regel om de bandbreedte te bepalen. Vooral Methode 3 is erg afhankelijk van de bandbreedte die gebruikt wordt. Wij wijten het mindere resultaat dan ook aan het feit dat de bandbreedte misschien niet helemaal optimaal gekozen is. Een andere manier om de bandbreedte bij methode 3 te verbeteren zou zijn om de datasets niet op te splitsen, en een kernschatter over de gehele dataset te nemen waar aan de datapunten gewichten toegekend worden. In theorie krijg je dan dezelfde kernschatter, maar de bandbreedte kan beter gekozen worden bij een grotere dataset.

8.3 Aantal datapunten

Een andere opmerking over de implementatie is dat het aantal gebruikte datapunten N niet constant is voor verschillende c. We maakten, voor zowel Methode 2 als 3, een slechte schatter met $\frac{N}{10}$ datapunten. Omdat er altijd een constant aantal datapunten gegenereerd werden (10.000), was het getal N sterk afhankelijk van c; als c groter was, was er een minder grote N, want de kans $\mathbb{P}[Y > c]$ was kleiner.

Het gevolg hiervan is dat het vergelijken van dezelfde methode voor verschillende c misschien geen eerlijke vergelijkingen zijn, maar dat verschillende methodes vergelijken voor dezelfde c nog wel altijd eerlijk is.

8.4 Limiet van aantal datapunten naar oneindig

Verder is het goed om te benoemen dat we ervan overtuigd zijn dat elke methode de ware verdeling aan zal nemen als het aantal datapunten, $N + M$, naar oneindig gaat.

8.5 Nadeel kernschatters

Zoals eerder in 2.4.5 is benoemd, hebben kernschatters een nadeel waar ze soms een gebied een positieve kans kunnen toekennen waar die eigenlijke waarde 0 zou moeten hebben. Dit probleem kan nog steeds een rol spelen bij de schatter die wij maken bij alle methodes omdat deze veelvuldig gebruik maken van kernschatters. Dit probleem zou eventueel verholpen kunnen worden door een lijn waarin gespiegeld moet worden te schatter, maar hier zijn wij niet verder op ingegaan wegens een gebrek aan tijd en omdat we geen aannames mochten maken van de verdeling in kwestie. Bij de exponentiële verdeling zou men deze lijn bijvoorbeeld schatter met $\frac{n}{n+1}$ keer de kleinste vernomen waarde.

8.6 Betrouwbaarheidsintervallen

In Figuren 6 en 7 valt het op dat de echte kansverdeling in het midden dichter bij de bovengrens van het betrouwbaarheidsinterval ligt, en links en rechts dichter bij de ondergrens. We kunnen hier geen reden voor geven. De betrouwbaarheidsintervallen zelf zijn puntschattingen, en dus geen functies die door onze methodes zijn gegenereerd. Toch zou het bekijken van de 80 individuele functies een hint kunnen geven naar waar dit verschijnsel vandaan komt.

Referenties

[1] Estimating distributions and densities, 23 November 2009. 36-350, Data Mining.

[2] David W. Scott. *Kernel Density Estimators*, chapter 6, pages 125–193. John Wiley & Sons, Ltd, 1992.

[3] S. J. Sheather and M. C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):683–690, 1991.

[4] webel od. Intro to kernel density estimation, 2018.

[5] Wikipedia bijdragers. Kernal density estimation — Wikipedia, the free encyclopedia, 2022. [Online; bezocht 30-Juni-2022].

A Script methode 1

```
import numpy as np
import scipy.stats
import scipy.integrate
import matplotlib.pyplot as plt
import openturns as ot

# Constants
n_samples = 10000
c = 1

# Parameters
mu_X = 0
mu_Y = 0
sigma_X_sq = 2
sigma_Y_sq = 1
cov = 0.5

# Generate data
data = np.random.multivariate_normal(np.array([mu_X, mu_Y]), np.matrix([[
    ↪ sigma_X_sq, cov], [cov, sigma_Y_sq]]), n_samples)

# Filter data and compute N, M
edge_case_data = data[data[:,1] > c]
X_where_Y_gt_c = edge_case_data[:,0]
N = len(edge_case_data)
M = n_samples - N

# P(Y > c)
edge_case_data_chance = N / float(N+M)

# Determine bad estimator
data_for_bad_estimation = data[:N // 10]
bad_estimator = scipy.stats.gaussian_kde(data_for_bad_estimation.T)

sample = ot.Sample([[v] for v in edge_case_data[:,1]])
bandwidth = ot.KernelSmoothing().computePluginBandwidth(sample)
f_Y_where_Y_gt_c_kernel_estimator = scipy.stats.gaussian_kde(edge_case_data
    ↪ [:,1], np.array(bandwidth).item())

def probability_X_le_x_given_y(x, y): #P[X <= x | Y = y] for y > c
    epsilon = 0.5 * 100 * (max(edge_case_data[:,1]) - c) / N #the constant 100 is
        ↪  how many points on average we use
    data_around_y = edge_case_data[(edge_case_data[:,1] > y - epsilon) & (
        ↪ edge_case_data[:,1] > y - epsilon)] #take points with Y-value at most
        ↪ epsilon away from y
    if len(data_around_y) == 0:
        #no data
        return 0
    data_around_y_X_le_x = data_around_y[data_around_y[:,0] <= x]
    return len(data_around_y_X_le_x) / len(data_around_y)
```

```
def F(x):
    integral_minus_inf_to_c = scipy.integrate.dblquad(lambda u, y: bad_estimator
        ↪ ([u, y]), -np.inf, c, -np.inf, x)[0]
    integral_minus_c_to_inf = scipy.integrate.quad(lambda y:
        ↪ probability_X_le_x_given_y(x, y) * (f_Y_where_Y_gt_c_kernel_estimator(
        ↪ y) + f_Y_where_Y_gt_c_kernel_estimator(2*c - y)) *
        ↪ edge_case_data_chance, c, np.inf)[0]
    return integral_minus_inf_to_c + integral_minus_c_to_inf

# Plotting
fig, axs = plt.subplots(3, constrained_layout=True)

X_min = data[:,0].min()
X_max = data[:,0].max()

x_axis_values = np.linspace(X_min, X_max, 1000)
F_x_values = np.array([F(x) for x in x_axis_values])
dx = x_axis_values[1] - x_axis_values[0]
f_x_values = np.gradient(F_x_values, dx)

axs[0].plot(x_axis_values, f_x_values)
axs[0].set_title("$f_X(x)$")

axs[1].plot(x_axis_values, F_x_values)
axs[1].set_title("$F_X(x)$")

axs[2].hist(data[:, 0], density=True, bins=np.arange(X_min, X_max, 0.2))
axs[2].set_title("Histogram␣of␣X␣(as␣density␣function)")

plt.show()
```

B Script methode 2

```
import numpy as np
import scipy.stats
import scipy.integrate
import matplotlib.pyplot as plt

# Constants
n_samples = 10000
c = 1

# Parameters
mu_X = 0
mu_Y = 0
sigma_X_sq = 2
sigma_Y_sq = 1
cov = 0.5

# Generate data
data = np.random.multivariate_normal(np.array([mu_X, mu_Y]), np.matrix([[
    ↪ sigma_X_sq, cov], [cov, sigma_Y_sq]]), n_samples)

# Filter data and compute N, M
edge_case_data = data[data[:,1] > c]
X_where_Y_gt_c = edge_case_data[:,0]
N = len(edge_case_data)
M = n_samples - N

# P(Y > c)
edge_case_data_chance = N / float(N+M)

# Determine bad estimator
data_for_bad_estimation = data[:N // 10]
bad_estimator = scipy.stats.gaussian_kde(data_for_bad_estimation.T)

def bad_estimator_x(x):
    return scipy.integrate.quad(lambda y: bad_estimator([x, y]), -np.inf, np.inf)
        ↪ [0]

# Kernel density estimator xy|y>c
f_XY_for_Y_gt_c_kernel_estimator = scipy.stats.gaussian_kde(edge_case_data.T)

def F(x):
    integral_minus_inf_to_c = scipy.integrate.dblquad(lambda u, y: bad_estimator
        ↪ ([u, y]), -np.inf, c, -np.inf, x)[0]
    integral_c_to_inf = scipy.integrate.dblquad(lambda u, y: (
        ↪ f_XY_for_Y_gt_c_kernel_estimator([u, y]) +
        ↪ f_XY_for_Y_gt_c_kernel_estimator([u, 2*c - y])) *
        ↪ edge_case_data_chance, c, np.inf, -np.inf, x)[0]
    return integral_minus_inf_to_c + integral_c_to_inf

# Plotting
fig, axs = plt.subplots(4, constrained_layout=True)
```

```
X_min = data[:,0].min()
X_max = data[:,0].max()

x_axis_values = np.linspace(X_min, X_max, 20)
F_x_values = np.array([F(x) for x in x_axis_values])
dx = x_axis_values[1] - x_axis_values[0]
f_x_values = np.gradient(F_x_values, dx)
bad_estimator_x_values = np.array([bad_estimator_x(x) for x in x_axis_values])

axs[0].plot(x_axis_values, f_x_values)
axs[0].set_title("$f_X(x)$")

axs[1].plot(x_axis_values, F_x_values)
axs[1].set_title("$F_X(x)$")

axs[2].plot(x_axis_values, bad_estimator_x_values)
axs[2].set_title("bad_estimator")

axs[3].hist(data[:, 0], density=True, bins=np.arange(X_min, X_max, 0.2))
axs[3].set_title("Histogram␣of␣X␣(as␣density␣function)")

plt.show()
```

C Script methode 3

```
import numpy as np
import scipy.stats
import scipy.integrate
import matplotlib.pyplot as plt

# Constants
n_samples = 10000
c = 1

# Parameters
mu_X = 0
mu_Y = 0
sigma_X_sq = 2
sigma_Y_sq = 1
cov = 0.5

# Generate data
data = np.random.multivariate_normal(np.array([mu_X, mu_Y]), np.matrix([[
    ↪ sigma_X_sq, cov], [cov, sigma_Y_sq]]), n_samples)
edge_case_data = data[data[:,1] > c]
N = len(edge_case_data)
data_for_bad_estimator = data[:N // 10]
M = n_samples - N
# P(Y > c)
edge_case_data_chance = N / float(N+M)

# Filter data and compute N, M
X_where_Y_gt_c = edge_case_data[:,0]
X_where_Y_le_c = data_for_bad_estimator[data_for_bad_estimator[:,1] <= c][:,0]

left_kde = scipy.stats.gaussian_kde(X_where_Y_le_c)
right_kde = scipy.stats.gaussian_kde(X_where_Y_gt_c)

def f(x):
    return (1 - edge_case_data_chance) * left_kde(x) + edge_case_data_chance *
        ↪ right_kde(x)

# Plotting
fig, axs = plt.subplots(2, constrained_layout=True)

X_min = data[:,0].min()
X_max = data[:,0].max()

x_axis_values = np.linspace(-6, 6, 300)
f_x_values = f(x_axis_values)

axs[0].plot(x_axis_values, f_x_values)
axs[0].plot(x_axis_values, scipy.stats.norm.pdf(x_axis_values, mu_X, np.sqrt(
    ↪ sigma_X_sq)))
axs[0].set_title("$f_X(x)$")
```

```
plt.show()
```