

Solvvenzkapital mit der LSMC-Methode:

Lineare Regression und neuronale Netze

Ralf Korn (RPTU Kaiserslautern-Landau, Fraunhofer ITWM, EI-QFM)

basierend auf gemeinsamen Arbeiten mit

- Anne-Sophie Krah (Generali, TU Kaiserslautern)
- Zoran Nikolic (Deloitte)

A.S. Krah, Z. Nikolić, R. Korn (2018) *A least-squares Monte Carlo framework in proxy modeling of life insurance companies*, Risks 6(2), doi:10.3390/risks6010001

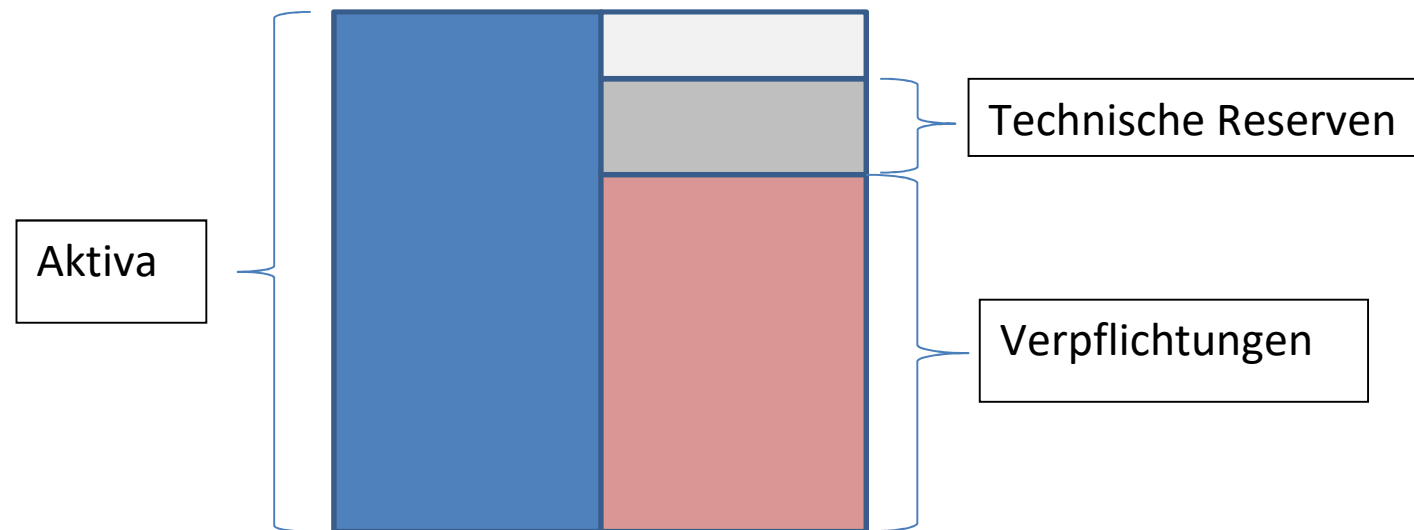
A.S. Krah, Z. Nikolić, R. Korn (2020) *Machine learning in least-squares Monte Carlo proxy modeling of life insurance companies*, Risks 8(1), doi:10.3390/risks80110021

A.S. Krah, Z. Nikolić, R. Korn (2020) *A least-squares Monte Carlo framework in proxy modeling of life insurance: Neural networks*, Risks 8(4).

1 Solvency II – Vorschriften für sichere Versicherer

Solvenzanforderung („Solvency Capital Requirement“ (SCR))

Jeder Versicherer muss zur Deckung seiner Verpflichtungen („Liabilities“) soviel Kapital zu Beginn des Jahres vorhalten, dass er **mit einer Wahrscheinlichkeit** von 99,5% am Ende des Jahres noch solvent ist.



1 Solvency II – Vorschriften für sichere Versicherer

Solvenzanforderung („Solvency Capital Requirement“ (SCR))

Jeder Versicherer muss zur Deckung seiner Verpflichtungen („Liabilities“) soviel Kapital zu Beginn des Jahres vorhalten, dass er **mit einer Wahrscheinlichkeit** von 99,5% am Ende des Jahres noch solvent ist.

European Council Directive 2009/138/EC

Article 122:

“Where practicable, insurance and reinsurance undertakings shall derive the solvency capital requirement directly from the **probability distribution forecast** generated by the internal model of those undertakings, using the Value-at-Risk measure set out in Article 101(3).”

- ⇒ Es ist die **komplette Verteilungsfunktion** vorherzusagen!
- ⇒ Es ist die **Verteilung der zufälligen Einflussfaktoren** zu simulieren und die der sich daraus ergebenden Differenz „**Verpflichtungen - Assets**“ der Versicherung !!!

1 Solvency II – Mögliche Berechnungskonzepte

Vorschläge:

- (Unrealistische/grob vereinfachte ?) Annahme für eine **Wahrscheinlichkeitsverteilung für die Verlustfunktion** (Normalverteilung, ...) => (vermutlich) keinerlei Akzeptanz bei den Regulatoren
- **Detaillierte MC Simulation** des **internen Modells**, um daraus eine hinreichend genaue empirische Verteilungsfunktion zu erzeugen => **vom rechnerischen Aufwand her nicht zulässig**
- Replizierungsansatz => Interessante Alternative, teils in der Praxis angewendet
- **Least-Squares Monte Carlo-Ansatz**

1 Solvency II-Berechnung (bei internen Modellen): Was brauchen wir dafür?

- **Modelle** für die (stochastische) Entwicklung der wesentlichen Einflussfaktoren für die Änderung der Werte der Assets und der Verpflichtungen des Versicherers
=> **Kapitalmarktmodelle, Mortalitätsmodelle, Stressszenarien ...**

⇒ Interne Modelle, GDV-Branchensimulationsmodell ...

- **Modelle**, wie sich Verpflichtungen aus Versicherungsverträgen ändern, wenn sich Umweltbedingungen (Kapitalmarkt, Sterblichkeit, Kundenverhalten, ...) ändern
=> **Cashflow-Modelle** für die einzelnen Versicherungsverträge ...

⇒ Cashflow-Projektionsmodell (CFPM)

- Cashflow-Modelle beinhalten Optionalitäten (Wählt ein Kunde die Verrentung oder sofortige Auszahlung,...) und Unsicherheiten z.B. bzgl. der (Kapital-) Marktentwicklung=> **Bewertung der sich aus dem CFPM ergebenden Zahlungsströme**

⇒ Optionsbewertungsproblem

Wir brauchen ein paar Exkurse

Exkurs 1 – Monte Carlo Methode

Monte Carlo Methode:

Berechne einen Erwartungswert $E(X)$ einer Zufallsvariablen durch wiederholte unabhängige Simulationen X_1, \dots, X_N der Zufallsvariablen („Experimente“) und approximiere

$$(MC) \quad E(X) \approx \frac{1}{N} \sum_{i=1}^N X_i =: \bar{X}_N$$

Beachte:

Falls $E(X)$ existiert konvergiert die MC-Methode aufgrund des **starken Gesetzes der großen Zahl** fast sicher.

Aufgrund des **zentralen Grenzwertsatzes** stellt (unter Zusatzbedingungen)

$$(MC-KI) \quad \left[\bar{X}_N - \frac{\sigma}{\sqrt{N}} q_{1-\alpha/2}, \bar{X}_N + \frac{\sigma}{\sqrt{N}} q_{1-\alpha/2} \right]$$

ein approximatives $(1-\alpha)$ -Konfidenzintervall für $E(X)$ dar, wobei q_z das z-Quantil der Standardnormalverteilung ist und $\sigma^2 = \text{Var}(X)$.

Exkurs 1 – Monte Carlo Methode

Beachte (Fortsetzung):

Wegen $q_{0,975} = 1.96$ ist die **2- σ -Regel**

$$(2\sigma) \quad \left[\bar{X}_N - 2 \frac{\hat{\sigma}}{\sqrt{N}}, \bar{X}_N + 2 \frac{\hat{\sigma}}{\sqrt{N}} \right], \quad \hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^N (X_i - \bar{X}_N)^2$$

als Approximation für das 95%-Konfidenzintervall populär.

Achtung:

Bisher klang alles gut („Alles konvergiert, Konfidenzintervalle werden klein ...“), aber die Länge des K-intervalls ist proportional zu $1/\sqrt{N}$

=> **Für jede weitere Stelle an Genauigkeit sind N^2 weitere Zufallsexperimente durchzuführen !!!**

=> MC-Methode ist **einfach und gut**, aber bei (hohen) Genauigkeitsanforderungen **sehr aufwändig**

Exkurs 2 – Cashflow-Projektionen (CFPM)

Hauptproblem: Zahlungsströme der Versicherung aus Prognosen berechnen

- Versicherer haben **umfangreiche Software**, die zu vorgegebenen ökonomischen Daten/Vorhersagen direkt auch zugehörige Zahlungsströme anhand der vorhandenen Verträge und Verpflichtungen berechnen kann, wobei diese noch mit Zufall behaftet sind (Cashflow-Projektionen).
- Versicherer haben auch einen großen Satz an **finanz- und versicherungsmathematischen Modellen**, mit deren Hilfe ökonomische Daten simuliert bzw. Vorhersagen erzeugt werden können (ESG=Economic Scenario Generator).

Klingt beides gut, aber:

Jeder einzelne Durchlauf einer Simulation und anschließender Umrechnung in einen einzigen entsprechenden Zahlungsstrom dauert bis zu einer Minute ...

=> Wollen wir MC-Methoden anwenden, muss es schneller gehen oder wir müssen mit wenigen Zahlungsströmen auskommen.

Exkurs 3 – Least-Squares Monte Carlo (LSMC)

Exkurs 3a – Lineare Regression und kleinste Quadrate

Aufgabe: Versuche einen funktionalen Zusammenhang $Y=f(X)$ zwischen Inputvariablen X und Output Y durch einen parametrischem Ansatz

$$f_{\beta}(x) = \beta_0 + \beta_1 x_1 + \dots + \beta_d x_d$$

zu schätzen, genauer:

Zu vorhandenen Daten- Paaren $(x^{(i)}, y^{(i)})$, $i=1, \dots, n (>d)$, $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, bestimme den **Kleinsten-Quadrate-Schätzer** $\hat{\beta}$ als Lösung von

$$(LS) \quad \min_{\beta \in \mathbb{R}^{d+1}} \sum_{i=1}^n \left(y^{(i)} - \beta_0 - \beta_1 x_1^{(i)} - \dots - \beta_d x_d^{(i)} \right)^2$$

Die Lösung ist explizit als

$$(LS\text{-Sch.}) \quad \hat{\beta} = (X^T X)^{-1} X^T Y$$

$$\text{mit } Y = (y^{(1)}, \dots, y^{(n)})^T \text{ und } X = \begin{pmatrix} 1 & x_1^{(1)} & \dots & x_d^{(1)} \\ \vdots & \vdots & & \vdots \\ 1 & x_1^{(n)} & \dots & x_d^{(n)} \end{pmatrix} \text{ gegeben.}$$

Exkurs 3a – Lineare Regression und kleinste Quadrate - 2

Ein unausrottbares Gerücht: *Mit linearer Regression erhält man lineare Approximationen an die gesuchte Funktion.*

Tatsächlich ist auch der Ansatz von Polynomen mit gemischten Monomen

$$f_{\beta}(x) = \sum_{i=0}^n \beta_i e_i(x)$$

wobei die Basisfunktionen $e_i(x)$ z.B. Monome mit gemischten Potenzen der Form

$$e_k(x) = \prod_{i=1}^d x_i^{j_i} \\ j_1 + \dots + j_d \leq k$$

sein können, eine lineare Regressionsfunktion, da sie **linear in den Koeffizienten** ist.

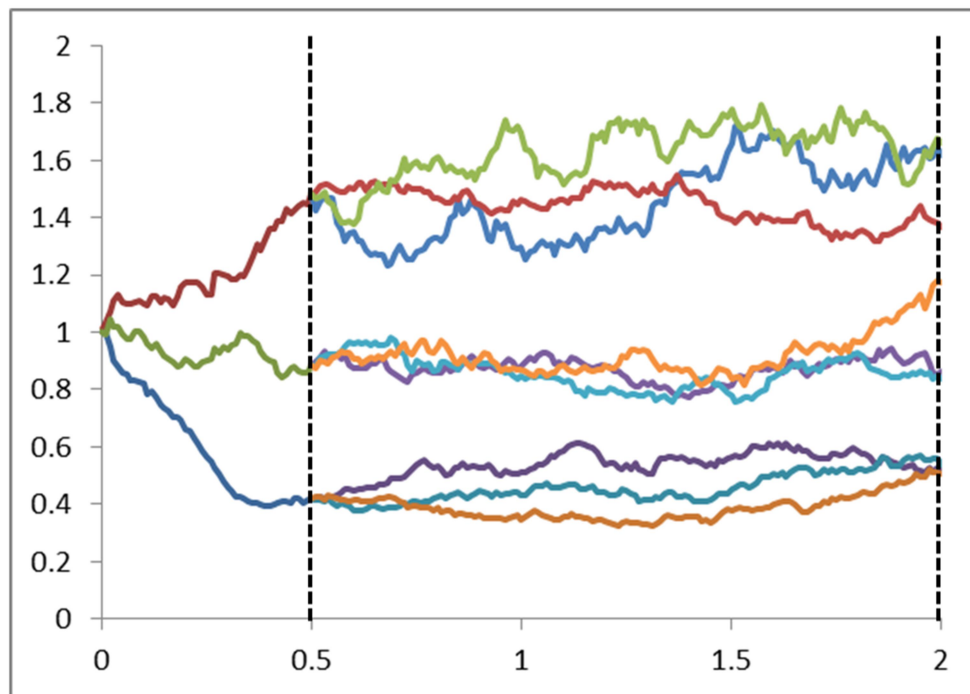
Exkurs 3 – Least-Squares Monte Carlo (LSMC)

Hauptidee: Schätze einen bedingten Erwartungswert durch Regression

Ein Beispiel: Bestimme die Verteilung eines Optionspreises $g(X(t))$ zur Zeit $t < T$ für eine Option mit Endzahlung $H = h(X(T))$ zur Zeit T

Problem: Es existiert **keine bekannte analytische Form** für $g(X(t))$

- Einfacher, genauer (aber zu aufwändiger) Ansatz: **Nested MC approach**



1. Simuliere eine große Anzahl, z.B. 100 000, Werte $X^{(i)}(t)$ zur Zeit t .
2. Simuliere für jedes $X^{(i)}(t)$ eine große Anzahl von Zahlungen $H^{(i,j)}$ gegeben $X^{(i)}(t)$. Berechne das Mittel $G(i)$ über die $H^{(i,j)}$ Werte als Approximation für die Bedingte Erwartung $E(H|X^{(i)}(t))$.
3. Approximiere die Verteilung von $g(X(t))$ durch die empirische Verteilung der $G(i)$.

Exkurs 3 – Least-Squares Monte Carlo (LSMC)

Hauptidee: Schätze einen bedingten Erwartungswert durch Regression

Ein Beispiel: Bestimme die Verteilung eines Optionspreises $g(X(t))$ zur Zeit $t < T$ für eine Option mit Endzahlung $H = h(X(T))$ zur Zeit T

Problem: Es existiert **keine bekannte analytische Form** für $g(X(t))$

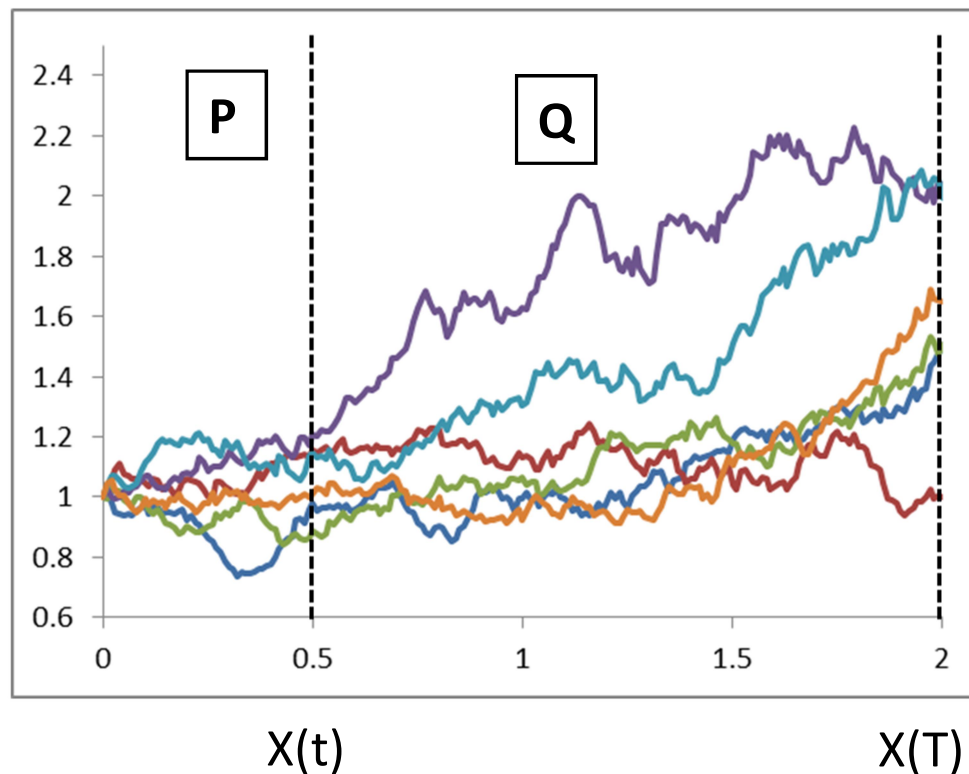
- Einfacher, genauer (aber zu aufwändiger) Ansatz: **Nested MC approach**
- **Least-squares MC:**
 - **Simuliere** hinreichend viele “Pfade” von $X(\cdot)$, die $X(t)$ und $X(T)$ beinhalten
 - **Berechne die Optionszahlung $h(X(T))$** für jeden “Pfad” von $X(\cdot)$, der $X(t)$ und $X(T)$ beinhaltet
 - Bestimme aus den Datenpaaren $(X(t), h(X(T)))$ eine **Regressionsfunktion** $r(X(t))$ als Approximation für $g(X(t))$
 - Simuliere eine große Zahl **repräsentativer Werte $X(t)$** , um eine **Approximation der Verteilungsfunktion** für $g(X(t))$ mittels der empirischen Verteilung der $r(X(t))$ zu erhalten

Exkurs 3 – Least-Squares Monte Carlo (LSMC)

Ein Beispiel: Bestimme die Verteilung eines Optionspreises $g(X(t))$ zur Zeit $t < T$ für eine Option mit Endzahlung $H = h(X(T))$ zur Zeit T

Problem: Es existiert keine bekannte analytische Form für $g(X(t))$

- **Least-squares MC:**



Beachte, dass wir jetzt den bedingten Erwartungswert $E(H|X^{(i)}(t))$ durch das “**Mittel**” über lediglich einen Wert “**approximieren**” !!!

Exkurs 3 – Least-Squares Monte Carlo (LSMC)

Hauptidee: Schätze einen bedingten Erwartungswert durch Regression

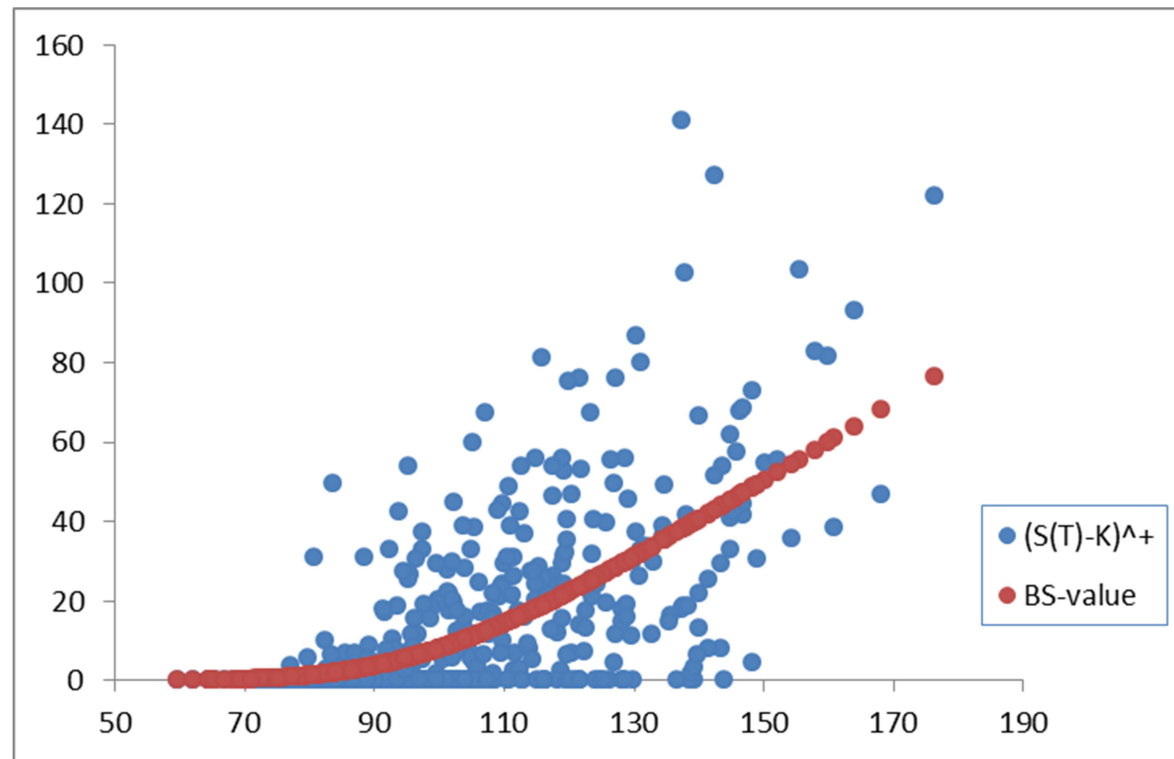
Frage: Warum braucht man die beiden Wahrscheinlichkeitsmaße P und Q , und warum gerade in der Reihenfolge „ P bis t , dann Q “ und nicht z.B. umgekehrt?

Antwort:

- P beschreibt die **Entwicklung der Welt** bis zum nächsten Bewertungszeitpunkt t („ P heißt **subjektives Maß**“)
- In t sind dann **zukünftige Zahlungen zu bewerten**, deshalb muss risiko-neutral simuliert werden, also unter Q („ Q heißt **risiko-neutrales Maß**“)

Exkurs 3: LSMC- Ein einfaches Optionsbewertungsbeispiel

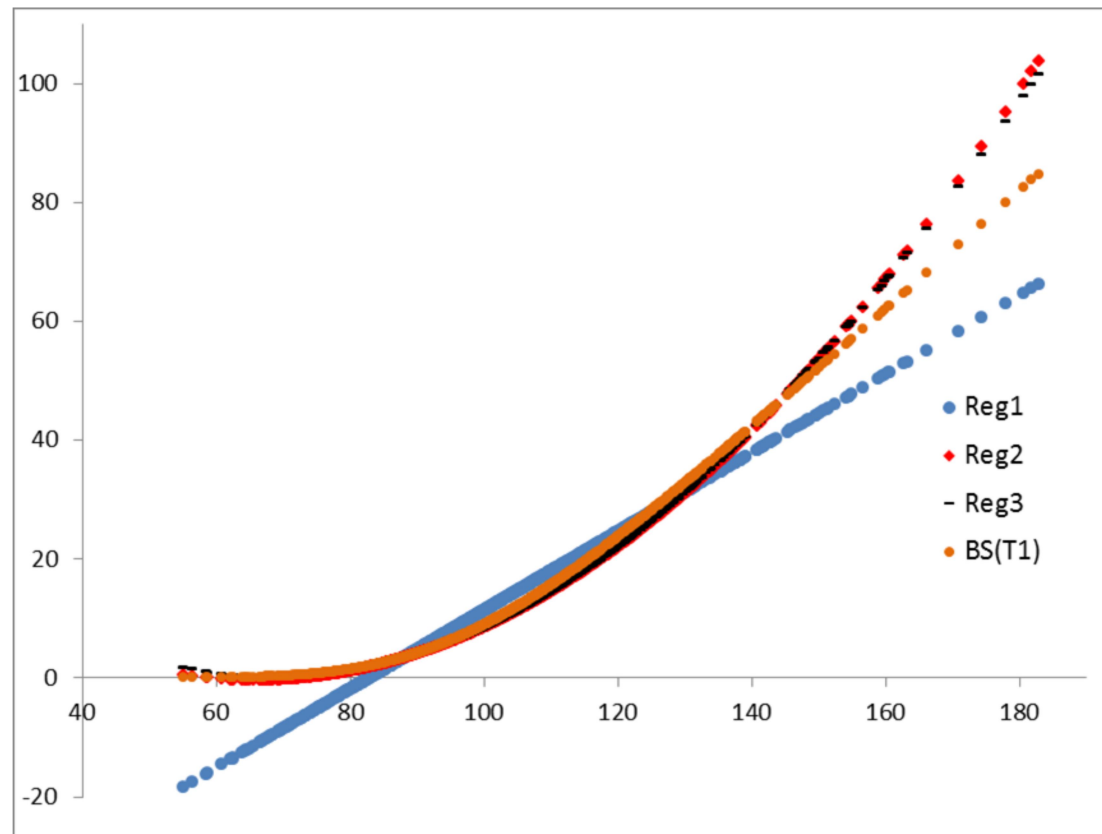
Europ. Calloption: Berechne den Preis $g(S(1))$ zur Zeit 1 einer Europäischen Calloption mit Endzahlung $H = (S(2)-100)^+$ und Fälligkeit 2, $S(0)=100$, mittels linearer, quadratischer und kubischer Regressionsfunktion (linear in den Parametern der Funktion!)



Simulierte Endzahlungen in T=2 (zu Werten in T=1) und zugehörige BS-Preise in T=1

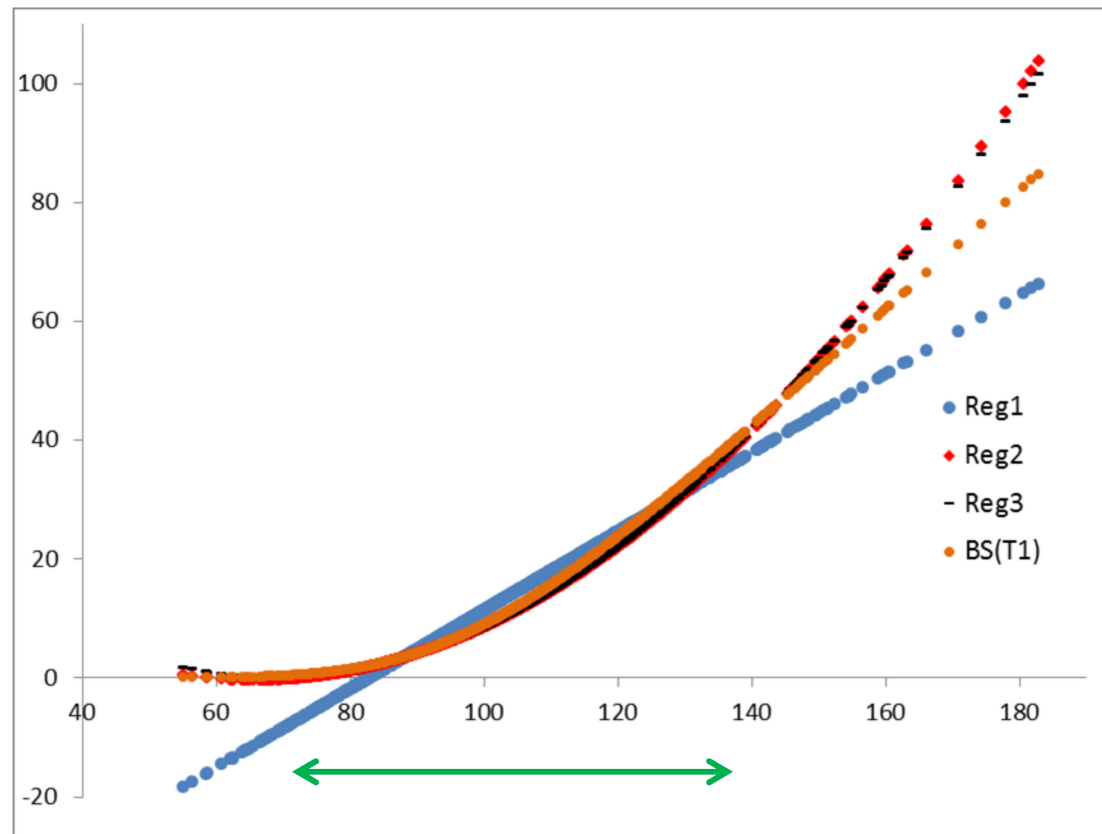
Exkurs 3: LSMC- Ein einfaches Optionsbewertungsbeispiel

Europ. Calloption: Berechne den Preis $g(S(1))$ zur Zeit 1 einer Europäischen Calloption mit Endzahlung $H = (S(2)-100)^+$ und Fälligkeit 2, $S(0)=100$, mittels linearer, quadratischer und kubischer Regressionsfunktion (linear in den Parametern der Funktion!)



Exkurs 3: LSMC- Ein einfaches Optionsbewertungsbeispiel

Europ. Calloption: Berechne den Preis $g(S(1))$ zur Zeit 1 einer Europäischen Calloption mit Endzahlung $H = (S(2)-100)^+$ und Fälligkeit 2, $S(0)=100$, mittels linearer, quadratischer und kubischer Regressionsfunktion (linear in den Parametern der Funktion!)



2 Least Squares Monte Carlo zur Berechnung des SCR: Einige Aspekte

Theoretische Rechtfertigung:

- Konvergenzresultate

Praktische Implementierung:

- Wahl der Regressionsfunktion
- Anzahl der Simulationsläufe zur Kalibrierung der Regressionsfunktion
- Wie wählt man die X-Werte?
- Wie erhält man die Y-Werte?
- Wie beurteilt man die Performance des Modells?
“Where to get the ground truth from?”

2 Least Squares Monte Carlo: Konvergenzresultate

Theoretische Begründung: Zwei Konvergenzaspekte (siehe Bauer/Ha (2018), Clément, Lamberton, Protter (2002)) => **Projektion und Monte Carlo**

Theorem

Es sei $F(X)=E(Y|X)$ ein Funktional von X in L^2 . Zu einer Menge von K linear unabhängigen Basisfunktionen $e_k(x)$ mit $e_0(x)=1$ sei die Projektion von $F(X)$ auf die Basisfunktionen

$$(1) \quad \hat{F}^{(K)}(X) = \sum_{k=0}^{K-1} \beta_k e_k(X)$$

und

$$(2) \quad \hat{F}^{(K,N)}(X) = \sum_{k=0}^{K-1} \hat{\beta}_k^{(N)} e_k(X)$$

ihre Approximation mit den kleinsten Quadrate-Schätzern mit N Realisierungen von X und Y .

Falls die Familie der Basisfunktionen in L^2 vollständig ist, so folgt:

$$(3) \quad \hat{F}^{(K)}(X) \xrightarrow{K \rightarrow \infty} F(X) \text{ in } L^2$$

$$(4) \quad \hat{F}^{(K,N)}(X) \xrightarrow{N \rightarrow \infty} \hat{F}^{(K)}(X) \text{ fast sicher}$$

2 Least Squares Monte Carlo zur Berechnung des SCR

Notwendige Schritte/Entscheidungen/Zutaten für den LSMC-Ansatz auf dem Weg zu einem zuverlässigen Approximationsverfahren für einen Lebensversicherer:

- eine detaillierte Beschreibung der Aufgabenstellung und der zu simulierenden Einfluss- und Zielgrößen
- ein Konzept für die Kalibrierung der Proxy-Funktion
- eine Validierungsprozedur für die erhaltene Proxy-Funktion
- die Anwendung des LSMC-Modells zur Vorhersage der Verlustverteilung

Notation: *Fitting Points* und *Fitting Values*

Ein Simulationslauf X unter dem subjektiven Maß P heißt ein **äußeres Szenario** oder **Fitting Point** (eine mögliche Entwicklung der (relevante) Welt über das Jahr).

Die durch die CFP-Methode (unter Q) erzeugten Projektionen von Y auf der Basis der simulierten Risikofaktoren $X = (X_1, \dots, X_d)$ heißen **innere Szenarien** oder **Fitting Values**.

3 LSMC-Proxy-Modellierung: Simulationsrahmen und Aufgabe

Die Aufgaben:

- Berechne die (volle) **Verlustverteilung** des Lebensversicherers zum Ende des Jahres
- Erhalte hieraus den SCR als das 99.5%-Quantil (der Differenz aus verfügbarem Kapital zu den Zeiten 1 und 0: $B_1 AC_1 - AC_0$)

Der Simulationsrahmen:

- Simuliere Realisierungen der **Risikofaktoren** X zur Zeit 1 unter P
- Berechne für jede Realisierung der Risikofaktoren das (abgezinste) verfügbare Kapital zur Zeit 1

$$AC(X) = E_Q \left(\sum_{t=1}^T B_t^{-1} Z_t | X \right) =: E_Q \left(\sum_{t=1}^T z_t(\phi_t(X)) | X \right)$$

wobei Z_t den Nettoprofit zur Zeit t und T das Ende des Projektionszeitraums bezeichnen. Beachte, dass wir jetzt simulieren, also die aktuelle CFP-Methode nutzen!

3 LSMC-Proxy-Modellierung: Simulationsrahmen und Aufgabe – 2

Ein Beispiel (siehe Table 1 in Krah, Nikolic, Korn (2018)):

Table 1. Risk factors in the CFP model.		
Component	Risk Factor Description	
X ₁	Risk-free interest rates movement	Kapitalmarktrisiken
X ₂	Change in interest rate volatility	
X ₃	Change in equity volatility	
X ₄	Shock on volatility adjustment (if used by the company)	
X ₅	Credit default	
X ₆	Credit spread widening	
X ₇	Currency exchange rate risk	
X ₈	Shock on equity market value	
X ₉	Shock on property market value	
X ₁₀	Lapse stress on best estimate assumptions	Aktuarielle Risiken
X ₁₁	Mortality catastrophe stress with a one-off increase in mortality	
X ₁₂	Mortality trend volatility stress	
X ₁₃	Mortality level stress on best estimate assumptions	
X ₁₄	Longevity trend volatility stress on best estimate assumptions	
X ₁₅	Longevity level stress on best estimate assumptions	
X ₁₆	Morbidity stress on best estimate assumptions	
X ₁₇	Expenses stress on best estimate assumptions	

3 LSMC-Proxy-Modellierung: Simulation äußerer und innerer Szenarien

Erzeuge die äußeren Szenarien (die **fitting points**) unter P, also

- zu einem stochast. Modell für jedes Risiko X_i simuliere Realisierungen $X^{(k)}$ von X ,
- verwende eine hohe Zahl äußerer Szenarien (oder eine Strategie, um den Wertebereich der Risiken abzudecken)

Erzeuge die zugehörigen inneren Szenarien (die **fitting values**) unter Q, also

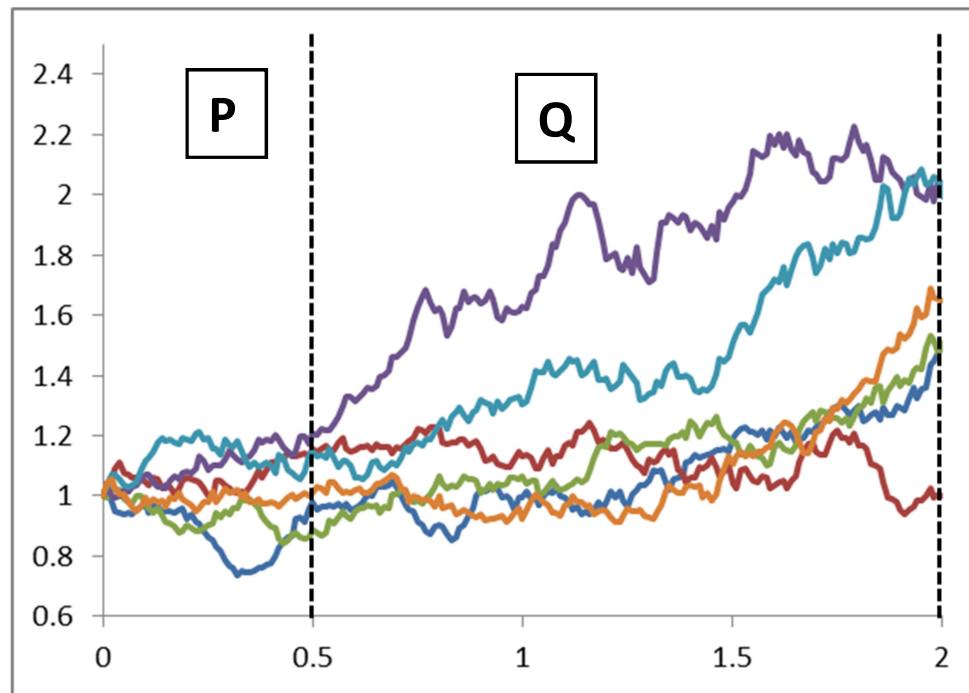
- verwende ein Cashflowprojektionstool (CFPM) zur Erzeugung weniger (idR 1 oder 2) marktkonsistenter Szenarien $\phi^{(k,j)}(X^{(k)})$ zu jedem äußeren Szenario
- erhalte die fitting values $Y^{(k)}$ via

$$Y^{(k)} = \frac{1}{a} \sum_{j=1}^a Y^{(k,j)} = \frac{1}{a} \sum_{j=1}^a \sum_{t=1}^T z_t \left(\phi^{(k,j)} \left(X^{(k)} \right) \right)$$

3 LSMC-Proxy-Modellierung: Simulation äußerer und innerer Szenarien

Nochmal zur Erinnerung

Äußere Szenarien Innere Szenarien



$X(t)$

Fitting Points

$X(T) \Rightarrow h(X(T))$

Fitting Values

3 LSMC-Proxy-Modellierung: Kalibrieren der Proxy-Funktion

Beachte: Paare $(X^{(k)}, Y^{(k)})$ zur Bestimmung einer Regressionsfunktion vorhanden

Frage: Wahl der Regressionsfunktion

Erster Vorschlag: Lineare Regression

Wähle (zusätzliche) Monome des Typs

$$e_k(x) = (x_1)^{j_1^k} \cdot \dots \cdot (x_d)^{j_d^k}$$

in **Schritt k** des Auswahlalgorithmus für die Monome als Basisfunktion.

Finde dann die optimalen Koeffizienten basierend auf den N fitting points and den fitting values, um die Proxy-Funktion durch Lösen von

$$\hat{\beta}^{(N)} = \arg \min_{\beta \in \mathbb{R}^K} \left\{ \sum_{i=1}^N \left(y^{(i)} - \sum_{k=0}^{K-1} \beta_k e_x(x^{(i)}) \right)^2 \right\}$$

zu erhalten.

3 LSMC-Proxy-Modellierung: Kalibrieren der Proxy-Funktion – 2

Ein Prinzip zur Wahl der Monome in Krah et al. (2018):

Prinzip der Marginalisierung

- Starte mit der konstanten Funktion
- Für mögliche Kandidaten für die nächste Basisfunktion müssen **all ihre partiellen Ableitungen** bereits als Basisfunktion ausgewählt worden sein
- Unter den Kandidaten wird der gewählt, der die größte Reduktion des AIC-Werts liefert (falls einer existiert und die Anzahl K_{\max} der Basisfunktionen noch nicht erreicht ist)

Ein Beispiel der Marginalisierung:

$x_1^2 x_2$ kann erst Kandidat werden, wenn $x_1^2, x_1 x_2, x_1, x_2$ schon gewählt wurden

3 LSMC-Proxy-Modellierung: Kalibrieren der Proxy-Funktion – 3

Erste 15 Iterationen in Krah et al. (2018)

Iteration	Exponents of Risk Factors														Regression Results		Out-of-Sample MSE	
k	$r_{k,1}$	$r_{k,2}$	$r_{k,3}$	$r_{k,4}$	$r_{k,5}$	$r_{k,6}$	$r_{k,7}$	$r_{k,8}$	$r_{k,9}$	$r_{k,10}$	$r_{k,11}$	$r_{k,12}$	$r_{k,13}$	$r_{k,14}$	$\hat{\beta}_k^{(N)}$	AIC	(1)	(2)
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8048.89	405,919.54	279,972.87	41,918.48
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	5796.16	347,121.57	22,744.22	32,479.31
2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	80.08	342,819.19	9863.42	5112.25
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	-28.03	337,759.88	4048.53	4413.39
4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	26.61	336,372.65	3003.24	4339.50
5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	108.73	335,245.70	2534.24	4329.35
6	0	0	0	0	0	0	0	0	0	0	0	1	0	0	22.68	334,903.08	2334.00	4314.37
7	0	0	0	1	0	0	0	1	0	0	0	0	0	0	-614.79	334,579.21	2257.41	4308.28
8	0	0	0	2	0	0	0	0	0	0	0	0	0	0	297.50	334,299.51	1341.89	1970.71
9	0	0	0	0	0	0	0	0	0	0	1	0	0	0	32.93	334,043.75	1376.76	1945.95
10	0	0	0	0	0	0	1	0	0	0	0	0	0	0	71.60	333,840.00	1084.23	1501.77
11	1	0	0	0	0	0	0	1	0	0	0	0	0	0	57.32	333,769.40	1134.52	1843.88
12	0	0	0	0	0	0	0	2	0	0	0	0	0	0	1636.91	333,523.57	638.87	748.50
13	0	1	0	0	0	0	0	1	0	0	0	0	0	0	-58.12	333,420.22	637.63	700.10
14	0	1	0	1	0	0	0	0	0	0	0	0	0	0	15.67	333,367.94	645.54	699.57
15	0	0	0	0	0	0	0	0	0	1	0	0	0	0	29.69	333,316.80	327.84	172.71

Nach 10 Iterationen (also $k=9$) wurden gewählt (in dieser Reihenfolge):

$$C, X_8, X_6, X_1, X_2, X_4, X_{12}, X_4 X_8, X_4^2, X_{11}$$

3 LSMC-Proxy-Modellierung: Kalibrieren der Proxy-Funktion – 4

Beachte: Paare $(X^{(k)}, Y^{(k)})$ zur Bestimmung einer Regressionsfunktion vorhanden

Frage: Wahl der Regressionsfunktion

Zweiter Vorschlag: Nicht-lineare Regression

in der Form eines Feedforward neuronalen Netzes (FNN)

$$f(x; \theta) = a_{L+1} \circ (g \circ a_L) \circ \dots \circ (g \circ a_1)(x),$$

$$a_l(x) = W(l)x + b(l),$$

mit

$f: \mathbb{R}^{N_0} \rightarrow \mathbb{R}^{N_{L+1}}, a_l: \mathbb{R}^{N_{l-1}} \rightarrow \mathbb{R}^{N_l}, g: \mathbb{R} \rightarrow \mathbb{R}$ ("Aktivierungsfunktion", Lipschitz f.)

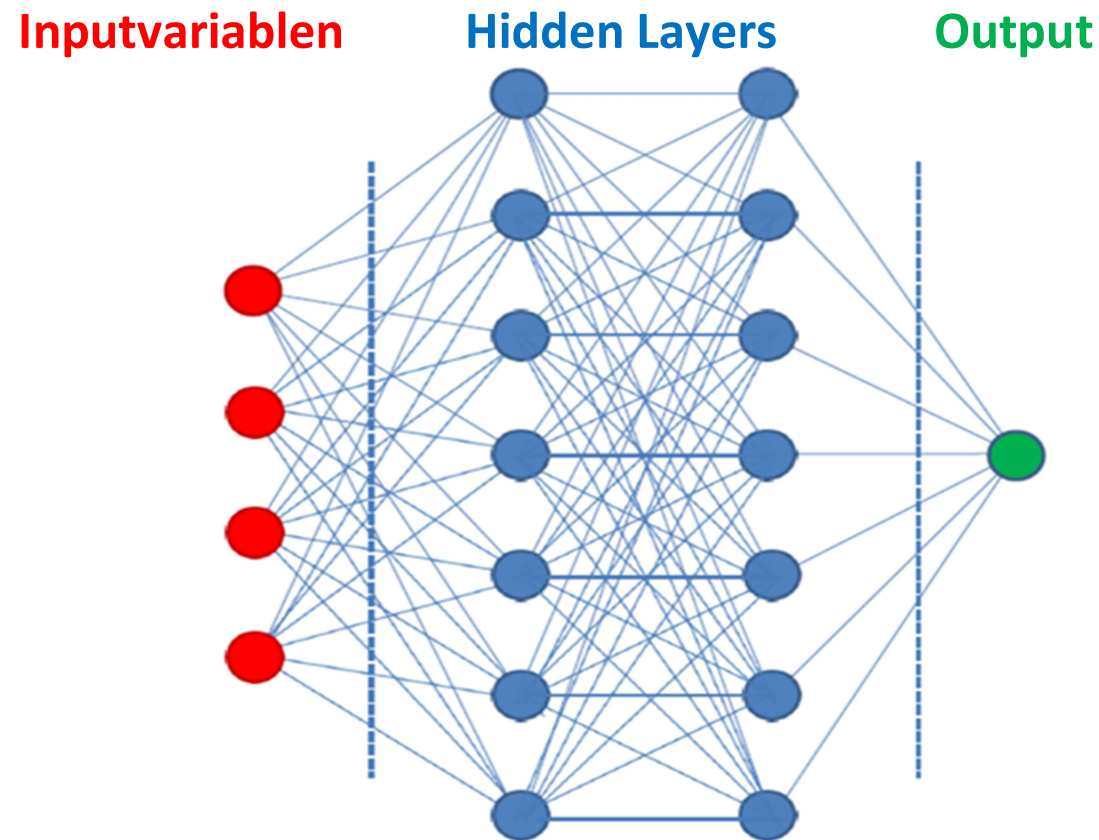
wobei g komponentenweise angewendet wird, also

$$(g \circ a_l)(z) = \left(g(W(l)_1 z + b(l)_1), \dots, g(W(l)_l z + b(l)_l) \right)^T$$

mit $\theta = (W, b)$, $W(l) \in \mathbb{R}^{N_l, N_{l-1}}, b(l) \in \mathbb{R}^{N_l}$, l der Zeilenindex.

3 LSMC-Proxy-Modellierung: Kalibrieren der Proxy-Funktion – 5

Ein vollständig verbundenes Feedforward neuronales Netz:



3 LSMC-Proxy-Modellierung: Kalibrieren der Proxy-Funktion – 6

Neuronale Netze benötigen eine andere Strategie der Kalibrierung

Hyperparameter Optimierung (HPO)

- Wähle die Form (“Architektur”) des FNN (Wieviele Neuronen? Wieviele verborgene Lagen ? ...)
- Wähle die Methode der Parameterkalibrierung des FNN (beinhaltet bereits viele Wahlen, Gradientenverfahren, Schrittweiten („Learning rates“), ...)
- (**Schon erledigt** Erster Schritte: Wähle den Typ des neuronalen Netzes ...)

Methoden der HPO

- Gittersuche (Probiere “jede Kombination” ...)
- Zufällige Gittersuche
- Bayessche Methoden

=> schon in der HPO-Phase werden NN an Teilmengen der Daten kalibriert/validiert.

4 LSMC-Proxy-Modellierung: Validieren der Proxy-Funktion

Bevor die Proxy-Funktion zur Simulation der Verteilung des verfügbaren Kapitals oder dem besten Schätzer der Verpflichtungen verwendet werden kann:

- Überprüfe, ob die Proxy-Funktion-Modellierung korrekte Werte liefert!

Validierungsstrategie:

- Erzeuge (oder wähle!) **einige wenige** (zwischen 15 und 200) äußere Szenarien $X^{(i)}$
- Erzeuge **viele** (zwischen 1000 und 16000) innere Szenarien pro äußerem Szenario (beachte die vertauschten Rollen !!!) um Validierungspunkte $Y^{(i)}$ zu erhalten
- Vergleiche die Validierungspunkte $Y^{(i)}$ mit den Werten, die die Anwendung der Proxy-Funktion auf $X^{(i)}$ liefert
- Ist die Validierung nicht zufriedenstellend, muss die Proxy-Funktion nachgebessert werden ...

Einige konkrete Out-of-Sample-Tests werden in Krah et al. (2018) beschrieben

5 LSMC-Proxy-Modellierung: Numerische Resultate

Anwendung der Proxy-Funktion

- Simuliere **viele** äußere Szenarien (genauer, 131072 „Real-World“-Szenarien)
- Setze ihre Resultate in die **Proxy-Funktion** ein, erhalte die jeweiligen Verluste
- **Ordne die Verluste an** und erhalte den SCR als das 99.5% Quantil

Das dauert nicht lange , da keine geschachtelten Simulationen oder CFP benötigt werden!

Bsp. in Krah et al. (2018) und (2020):

- 14 Risikofaktoren, 25.000 fitting points mit je $a=2$ inneren Szenarien per äußerem

Lineare Regression:

- Proxy-Funktion mit **60 (bzw. 224) Monomen**
- Validierung auf zwei Validierungsmengen erfolgreich passiert
- Genauigkeit gut, gesamte Rechenzeit < 6 Stunden auf einem Cluster, während für geschachtelte Simulation mindestens 25 Wochen geschätzt werden

5 LSMC-Proxy-Modellierung: Numerische Resultate – 2

Ergebnisse für Neuronale Netze:

Es wurden 300 FNN durch Variation der folgenden Parameter/Methoden mittels Sobol-Gittersuche trainiert (Bezeichnungen unten gemäß KERAS)

1. Anzahl Hidden Layer: zwischen 2 und 10
2. Anzahl Neuronen pro Hidden Layer: zw. 16 und 128 (für alle Layer gleich)
3. Aktivierungsfunktionen: *Sigmoid*, *ReLU*, *LeakyReLU* (für alle Layer gleich)
4. Akt.fkt. für den Output: Sigmoid, Linear
5. Optimierer: *Nadam*, *Adam*, *Adamax*
6. Lernrate (Schrittweite): zwischen 0,0005 und 0,005
7. Dropoutrate: zwischen 0 und 0,4
8. Anfangswertwahl: *GlorotUniform*, *RandomUniform*, *RandomNormal*
9. BatchSize: 100, 200, 400, 800, 1600

Beachte, dass es hier deutlich mehr mögliche obige Kombinationen als „nur“ 300 gibt

...

5 LSMC-Proxy-Modellierung: Numerische Resultate – 3

Ergebnisse für Neuronale Netze:

Aus den 10 besten neuronalen Netzen, also den folgenden (siehe Krah et al. (2020))

Table 1. Hyperparameter combinations producing the 10 best performing neural networks.

ID	Hidd. layers	Neurons	Activ. func.	Activ. outp.	Optimizer	Learning rate	Dropout rate	Initializer	Batch size
NN1	4	80	Sigmoid	Linear	Nadam	0.003488281	0.003125	GlorotUniform	1600
NN2	6	67	Sigmoid	Linear	Nadam	0.004929688	0.00625	GlorotUniform	400
NN3	3	114	Sigmoid	Sigmoid	Adamax	0.0044375	0.05	GlorotUniform	800
NN4	6	95	Sigmoid	Linear	Adam	0.002662109	0.0390625	GlorotUniform	400
NN5	4	63	Sigmoid	Sigmoid	Adamax	0.001976563	0.06875	RandomNormal	400
NN6	5	44	Sigmoid	Linear	Adam	0.003356445	0.00078125	RandomUniform	1600
NN7	6	101	Sigmoid	Sigmoid	Adamax	0.004332031	0.028125	RandomNormal	400
NN8	6	126	Sigmoid	Linear	Adamax	0.003365234	0.0515625	GlorotUniform	1600
NN9	3	113	Sigmoid	Sigmoid	Adamax	0.003426758	0.14453125	GlorotUniform	800
NN10	5	104	Sigmoid	Sigmoid	Adam	0.001220703	0.1046875	RandomNormal	400

konstruieren wir ein **Ensemble** (d.h. wir verwenden die durchschnittliche Vorhersage der 10 besten neuronalen Netze).

5 LSMC-Proxy-Modellierung: Numerische Resultate – 4

Ensemble Neuronales Netz vs. 224-Monome lineares Regressionspolynomial

(Tabelle aus Krah et al. (2020))

Table 3. Validation figures for 300-886 AIC OLS polynomial and the ensemble of 10 best performing neural networks after the exclusion of the points with scenarios outside of the fitting space. MAEs in %.

v.mae	v.mae ^a	v.res	v.mae ⁰	v.res ⁰	ns.mae	ns.mae ^a	ns.res	ns.mae ⁰	ns.res ⁰	cr.mae	cr.mae ^a	cr.res	cr.mae ⁰	cr.res ⁰
300-886 AIC OLS polynomial with 224 terms														
0.194	0.186	−8.7	5.329	19.9	0.253	0.244	−34.6	3.608	−6.0	0.124	0.121	−10.3	4.159	18.4
Ensemble of the best 10 neural networks														
0.091	0.087	−1.8	2.583	−9.8	0.067	0.065	3.1	2.292	−5.0	0.091	0.090	7.7	2.350	−0.4

Das Ensemble NN schlägt das OLS-Polynom klar auf der Validierungsmenge.

Aber: Das muss nicht das Ende der Optimierungen sein

5 LSMC-Proxy-Modellierung: Numerische Resultate – 5

Sensitivitätsanalyse und Interpretierbarkeit (siehe Krah et al. (2020))

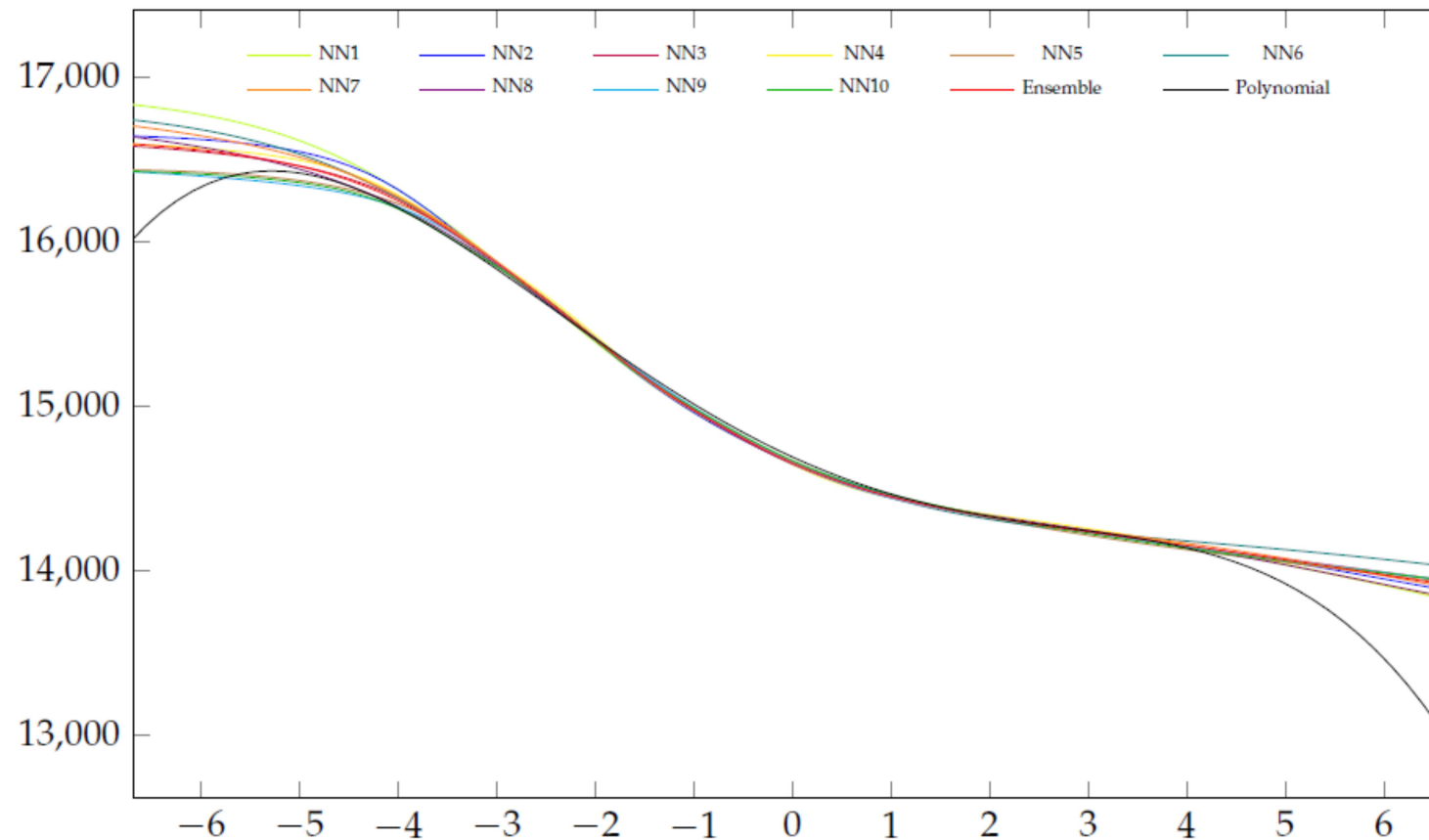


Figure A3. One-dimensional curves of the 10 best performing neural networks, their ensemble and the polynomial with respect to X_1 .

5 LSMC-Proxy-Modellierung: Numerische Resultate – 6

Sensitivitätsanalyse und Interpretierbarkeit (siehe Krah et al. (2020))

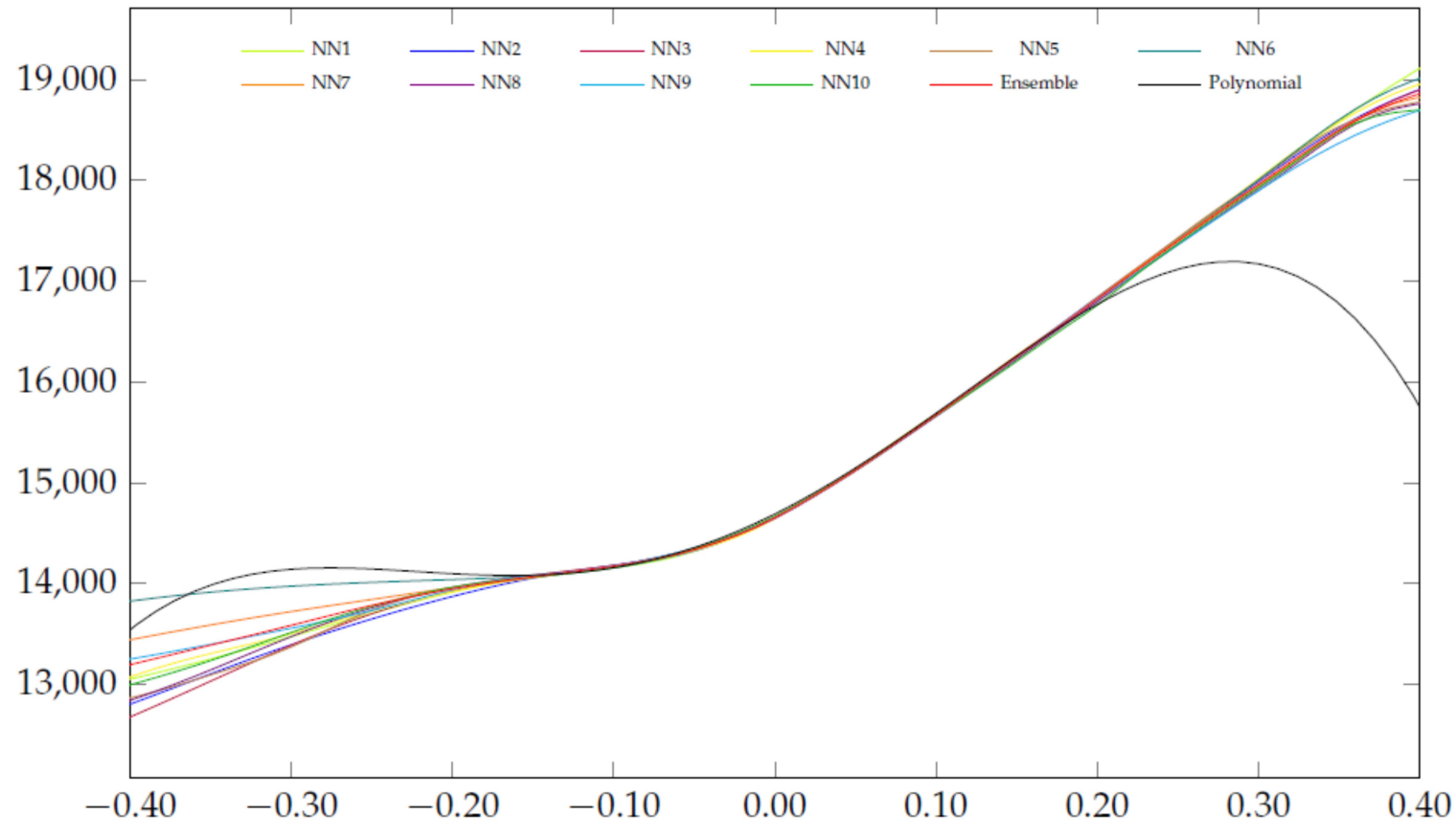


Figure A4. One-dimensional curves of the 10 best performing neural networks, their ensemble and the polynomial with respect to X_8 .

5 LSMC-Proxy-Modellierung: Numerische Resultate – 7

Sensitivitätsanalyse und Interpretierbarkeit (siehe Krah et al. (2020))

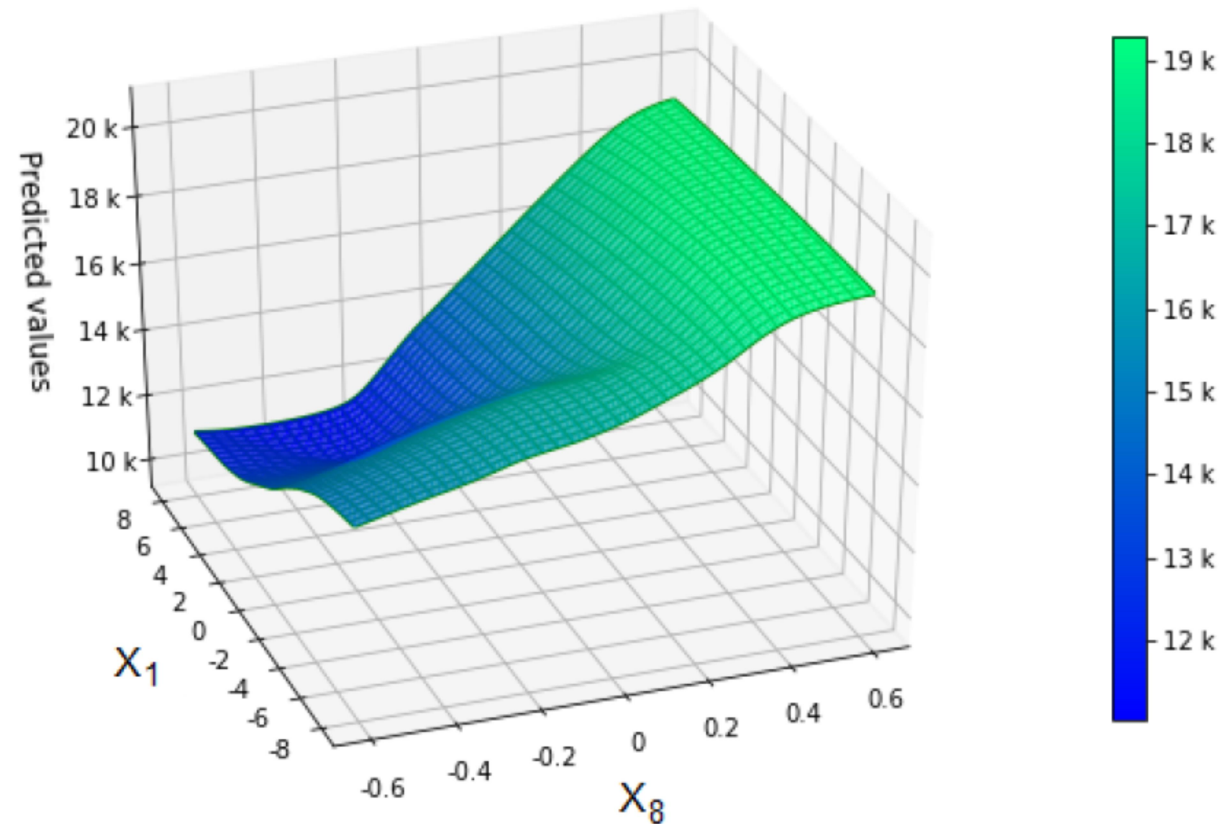


Figure A5. Two-dimensional curve of the ensemble of 10 best performing neural networks with respect to X_1 and X_8 .

5 LSMC-Proxy-Modellierung: Numerische Resultate – 8

Sensitivitätsanalyse und Interpretierbarkeit (siehe Krah et al. (2020))

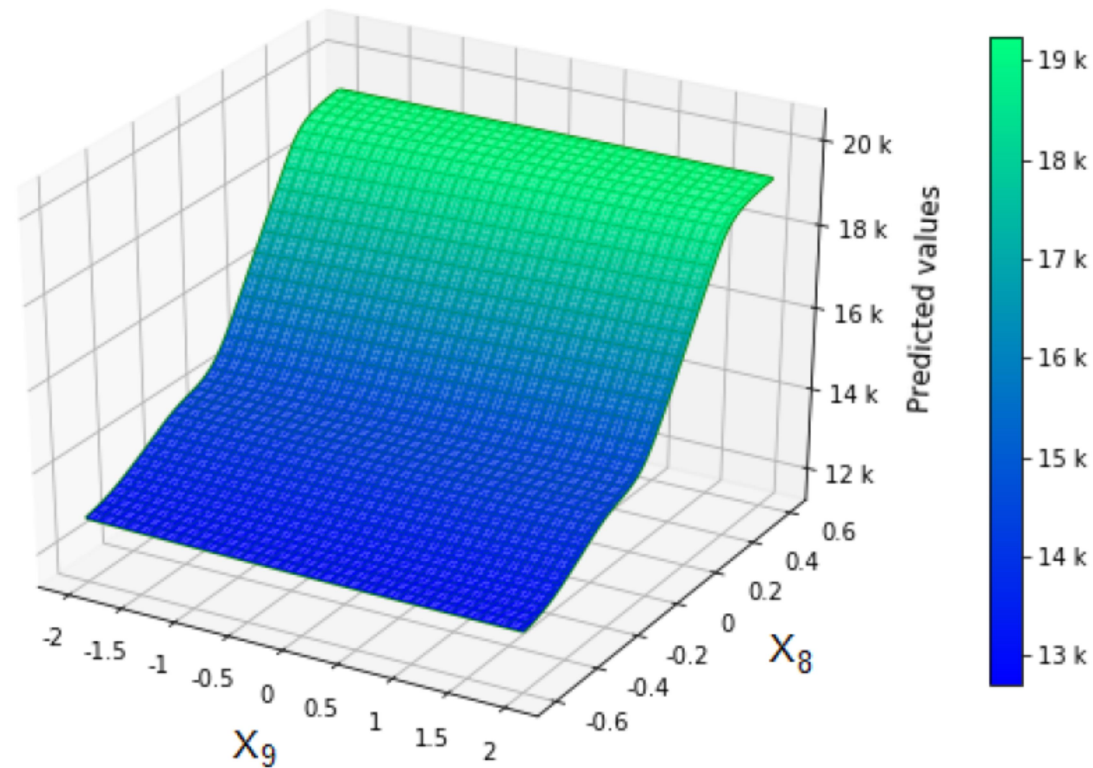


Figure A6. Two-dimensional curve of the ensemble of 10 best performing neural networks with respect to X_8 and X_9 .

5 LSMC-Proxy Modellierung: Visionen, Fragen und Wünsche

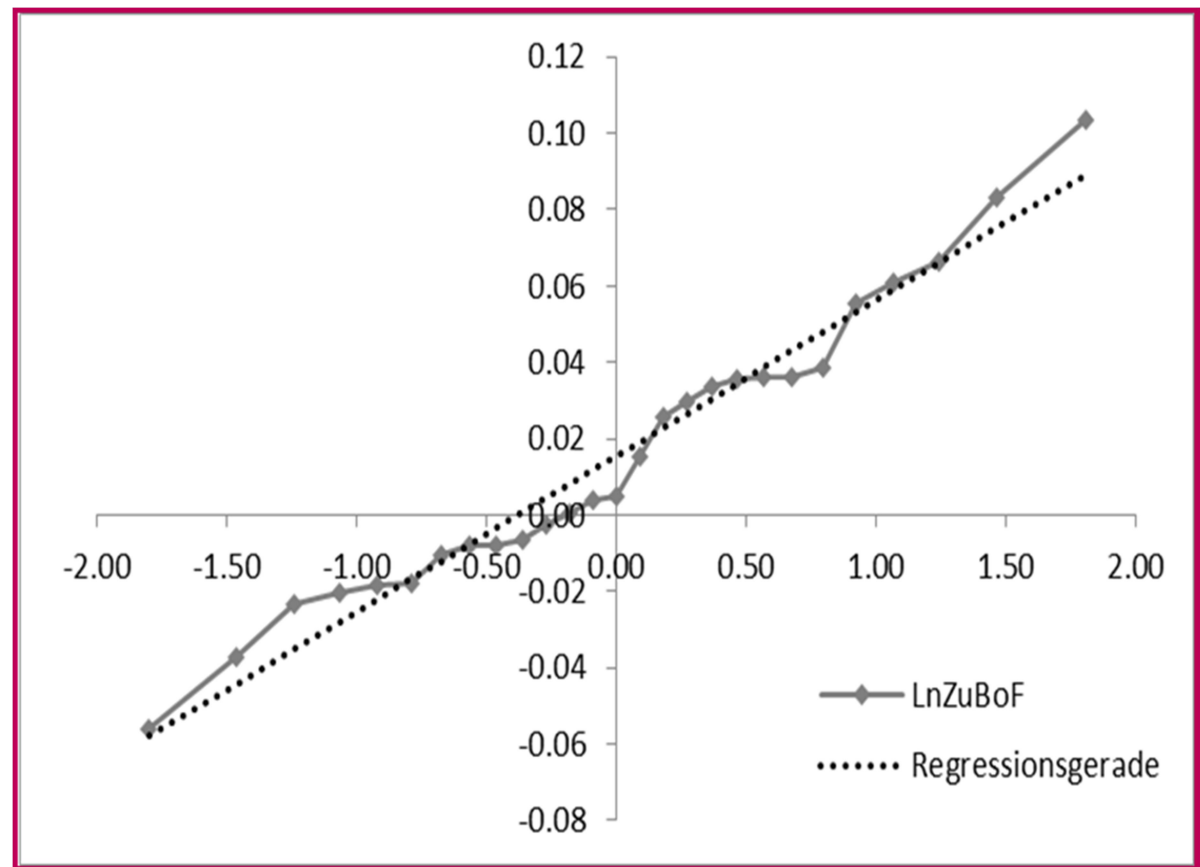
- Wie erhält man das “bestmögliche” Netzwerk, die beste Architektur, ...
- Weiß man was über Fehlerschranken?
- Lassen sich Strukturen/Beziehungen einbauen, die durch finanz- oder versicherungsmathematische Überlegungen gelten sollen?
- Können wir auch das Cashflowprojektionstool lernen (=> deutliche ambitionierter!!!)
- **Haben wir genug Daten für Deep Learning?**

6 Ein Ausblick: Sind Normalverteilungsannahmen akzeptabel?

Normal-QQ-Plot für die log-BoF-Zuwächse 2015-2022 (großer deutscher Versicherer)

(aus R. Korn, G. Stahl „Insights from Backtesting Solvency II“)

- **Daten:**
Quartalsdaten 2015Q4 – 2022Q1 SCR, Basic Own Funds, Aktienkurse, ...
- Normalverteilungsapproximationen für die Own Funds und für den CAR („Solvenzkapitalquotient“) sind adäquat
- R^2 -Werte oberhalb 0.9



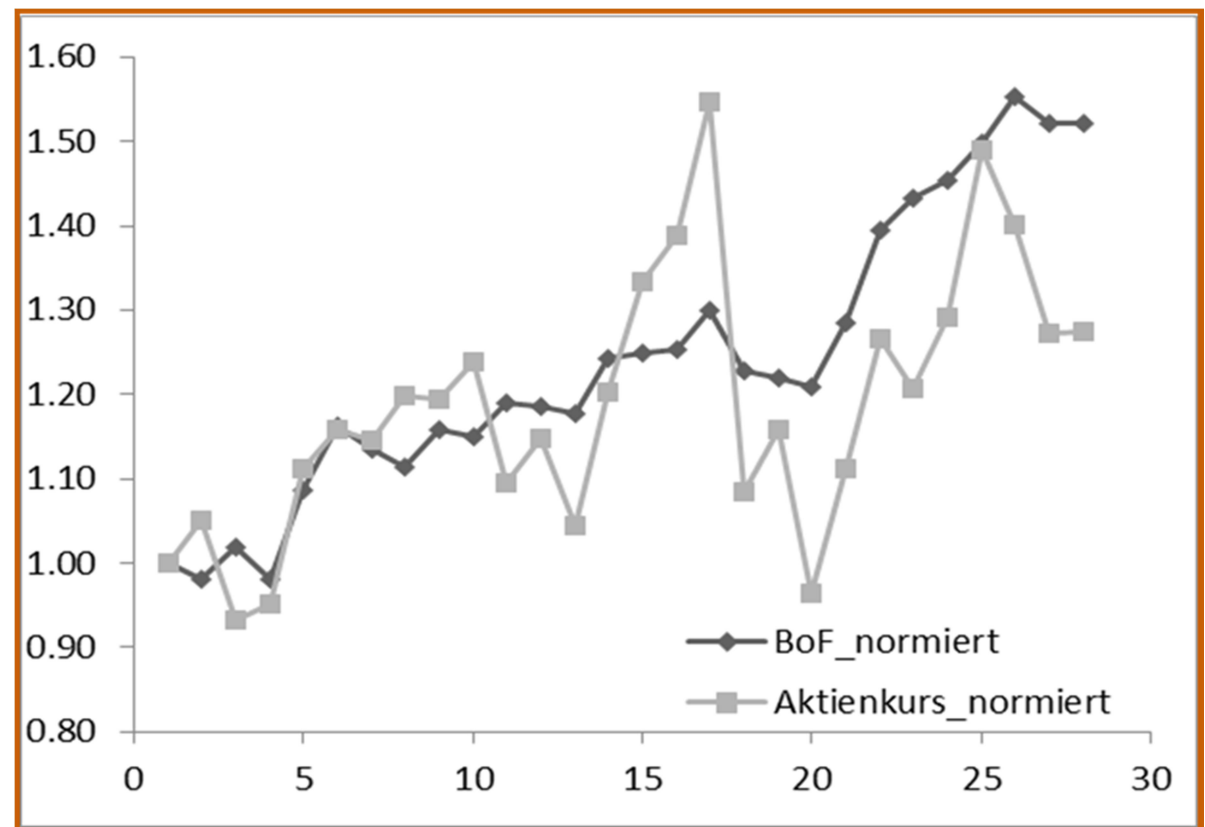
6 Ein Ausblick: Sind Normalverteilungsannahmen akzeptabel -2

- Aktienpreisvolatilität ist deutlich höher als die Volatilität der Own Funds
- Der SCR wird deutlich überschätzt
- Mit den Daten aus dem vorangegangenen Beispiel ist der SCR **um** einen **Faktor 3** zu groß

Volatilitäten	
Log_Aktie	0.234441209
Log_BoF	0.075437655

Normierte Werte der Basic Own Funds und des Aktienkurses (großer deutscher Versicherer)

(aus R. Korn, G. Stahl „Insights from Backtesting Solvency II“)



Referenzen

D. Bauer, H. Ha (2018) *A least-squares Monte Carlo approach to the calculation of capital requirements*. Working paper.

C. Bettels, J. Fabrega, C. Weiß (2014) *Anwendung von Least Squares Monte Carlo (LSMC) im Solvency-II-Kontext—Teil 1*. Der Aktuar 2: 85–91.

M. Broadie, Y. Du, C. Moallemi (2015) *Risk estimation via regression*. Operations Research, 63, 1077–1097.

J. Carriere (1996) *Valuation of Early-Exercise Price of Options Using Simulations and Nonparametric Regression*. Insurance: Mathematics and Economics, 19, 19–30.

E. Clément, D. Lamberton, P. Protter (2002) *An analysis of a least squares regression method for American option pricing*. Finance and Stochastics, 6, 449–471.

P. Glasserman, B. Yu (2004) *Simulation for American options: regression now or regression later?* Monte Carlo and Quasi-Monte Carlo Methods 2002, 213–226, Springer.

R. Korn (2014). *Moderne Finanzmathematik - Theorie und praktische Anwendungen*. Band I. Springer.

R. Korn, E. Korn, G. Kroisandt (2010) *Monte Carlo Methods and Models in Finance and Insurance*. Chapman & Hall, CRC-Finance Series.

R. Korn, G. Stahl (2023) *Insights from Backtesting Solvency II*, submitted.

A.S. Krah, Z. Nikolić, R. Korn (2018) *A least-squares Monte Carlo framework in proxy modeling of life insurance companies*, Risks 6(2), doi:10.3390/risks6010001

A.S. Krah, Z. Nikolić, R. Korn (2020) *Machine learning in least-squares Monte Carlo proxy modeling of life insurance companies*, Risks 8(1), doi:10.3390/risks80110021

A.S. Krah, Z. Nikolić, R. Korn (2020) *A least-squares Monte Carlo framework in proxy modeling of life insurance: Neural networks*, Risks 8(4).

Z. Nikolić, C. Jonen, C. Zhu (2017) *Robust regression technique in LSMC proxy modeling*. Der Aktuar 1: 8–16.