



UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR SOFTWARETECHNIK
UND PROGRAMMIERSPRACHEN

Stochastisches Routing

Stochastic Routing

Bachelorarbeit

im Rahmen des Studiengangs
Informatik
der Universität zu Lübeck

vorgelegt von
Max Lorenz

ausgegeben und betreut von
Prof. Dr. Leucker

mit Unterstützung von
René Schönfelder

Lübeck, den 4. Juni 2015

Abstract

Im Folgenden wird ein alternativer Ansatz für Routing unter Verwendung statistischer Methoden entworfen und eine Evaluation beschrieben. Dabei wird auf die Frage eingegangen, wie man Routing verändern muss, um nicht nach der Strecke oder der Zeit, sondern nach Energieverbrauch oder jedem anderen stochastischem Prozess optimieren zu können. Um dies zu tun, wird ein stochastische Modell vorgestellt, welches sich am Besten für die Aufgabe eignet ein Algorithmus, der das Routing mit diesem Modell erlaubt, entworfen.

In diesem Rahmen wird zu dem Ergebnis gekommen, dass die Repräsentation der einzelnen Kantengewichte durch diskrete Werte alle Anforderungen erfüllt. Mit dem hier vorgestellten Modell ist es möglich, mit einem vernünftigen Zeitaufwand ein Routing zu implementieren, welches nach dem Akkustand optimieren kann und maximal flexibel ist.

Ein offensichtliches Einsatzgebiet ist in Navigationssoftware, der Algorithmus ist jedoch in vielen weiteren Gebieten einsetzbar, in denen es darum geht, stochastisch abhängige Graphen zu untersuchen. Dies ist zum Beispiel in der Verkehrsplanung der Fall.

Erklärung

Ich versichere an Eides statt, die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Hilfsmittel angefertigt zu haben.

Lübeck, 4. Juni 2015

Inhaltsverzeichnis

1	Einleitung	1
1.1	Fragestellung	2
1.2	Begriffsabgrenzung	3
1.3	Wissensstand	3
1.4	Ausgangssituation	4
1.5	Problembeschreibung	4
1.6	Ziel meiner Arbeit	6
2	Lösungsweg	7
2.1	Routing	7
2.1.1	Stochastische Kostenfunktion	8
2.2	Verteilungsfunktion und Wahrscheinlichkeitsfunktion	8
2.3	Anforderungen an eine stochastische Kostenfunktion	11
2.3.1	Faltung von Verteilungen	12
2.4	Arten von Verteilungen	12
2.4.1	Alpha-stabile Verteilungen	12
2.5	Diskrete Verteilungen	18
2.5.1	Definition	19
2.5.2	Beschreibungskomplexität	19
2.5.3	Beispiel	20
3	Ergebnisse	23
3.1	Vergleichskriterien	23
3.1.1	Vergleich der Form	23
3.1.2	Vergleich der mathematischen Eigenschaften	25

4	Lösung	27
4.1	Spezifikation	27
4.1.1	Reichweitenoptimierung	27
4.1.2	Fahrzeitminimierung	27
4.2	Algorithmische Komplexität	28
4.2.1	Summenformel	28
4.2.2	Mit Nutzung der FFT	29
5	Schlussfolgerung	31
5.1	Evaluation	31
5.1.1	Versuchsaufbau für eine mögliche Evaluation	31
5.2	Vorteile	32
5.2.1	Modellbildung	32
5.2.2	Adaption	32
5.2.3	Echtzeitfähigkeit	33
5.3	Nachteile	33
5.3.1	Rundungsfehler	33
5.3.2	Mathematische Eleganz	33

1 Einleitung

Navigationsgeräte sind aus unserem Alltag nicht mehr wegzudenken. Sie können uns die kürzeste oder die schnellste Route zwischen zwei Punkten berechnen. Der Algorithmus für diese Berechnung ist in beiden Fällen der gleiche: die Strecke zwischen zwei Punkten lässt sich als Skalar darstellen und je nachdem, ob wir die Dauer der Fahrt oder die Distanz betrachten, können wir nun den kürzesten Weg durch den erzeugten Graphen berechnen. Hierfür kommt zumeist eine optimierte Variante von Dijkstras Algorithmus zum Einsatz.

Auf den ersten Blick scheint dieses Modell gute Ergebnisse zu liefern. Es gibt jedoch Phänomene, die sich damit nicht abbilden und somit nicht berechnen lassen. Ein Beispiel:

Angenommen, wir haben eine Strecke, die mit einer Wahrscheinlichkeit von fünfzig Prozent in einer Minute geschafft werden kann, und mit einer Wahrscheinlichkeit von fünfzig Prozent 99 Minuten dauert. Dies ergibt einen Erwartungswert von 50 Minuten. Dieser Wert ist ein Skalar, der sich als Parameter für die Berechnung eignet, er gibt jedoch niemals die Realität wieder. Problematisch kann das werden, wenn wir ein Elektrofahrzeug mit einer beschränkten Reichweite betrachten.

Autohersteller haben bereits damit begonnen, nach geeigneten Lösungen zu suchen. So ist im neuen BMW i3 ein Navigationsgerät verbaut, welches unter dem Namen *BMW i Navigation* läuft und einen Reichweitenassistenten bietet. Auch von *Pioneer*, dem Japanischen Elektronikkonzern, wurde ein vergleichbares Produkt präsentiert, welches bisher (Stand April 2015) allerdings nur in Japan vertrieben wird.

Welche Algorithmen hinter den jeweiligen System stehen, ist nicht bekannt. Da es im

1 Einleitung

Gegensatz zum klassischen Routing, welches in traditionellen Navigationsgeräten zum Einsatz kommt, keinen gängigen Algorithmus gibt, möchte ich in meiner Arbeit untersuchen, welche Datenstrukturen und Algorithmen sich dafür eignen.

Das Routing mit Graphen, in denen die Kanten die Entfernung oder die benötigte Zeit, die Zwischen zwei Knoten liegt, beschreiben, ist bereits vielfach in der Fachliteratur beschrieben worden. Ein Verfahren, welches die Wahrscheinlichkeit maximiert, mit der man vom Start eines Graphen zu dem Ziel gelangt oder welches die kürzeste Strecke mit einer Erreichenswahrscheinlichkeit von über 99 Prozent ermittelt, gibt es nur mit Einschränkungen. Außerdem sind diese Verfahren oft an eine bestimmte Domäne gebunden und nicht allgemein genug, um damit generelles stochastisches Routing betreiben zu können. Im Verlauf meiner Arbeit werde ich zeigen, dass diese Einschränkungen durch die geschickte Wahl eines Modells umgangen werden können. Dies ermöglicht die Entwicklung eines Algorithmus, welcher so allgemein ist, dass er für eine Vielzahl von Problemen anwendbar ist.

1.1 Fragestellung

Wie man leicht erkennt, ist das Thema nicht klar eingegrenzt und besitzt zu viele Aspekte, um diese alle in einer Bachelorarbeit abdecken zu können. Daher werde ich in dieser Arbeit nicht näher darauf eingehen, wie man den Akkuverbrauch einer Strecke modellieren oder die Stauwahrscheinlichkeiten dieser beschreiben kann. Ich möchte mich darauf konzentrieren, ein Modell zu finden, welches eine allgemeingültige Beschreibung aller dieser Anwendungsgebiete ermöglicht. Dabei möchte ich allerdings darauf achten, dass sich dieses auch in der Praxis bewährt und aufzeigen, wie eine mögliche Implementierung aussehen und evaluiert werden kann.

Zu dem Thema gehört auch die Fragestellung, wie ein Routing mit so einem Modell aussieht und welche Vergleichskriterien für stochastische Beschreibungen sinnvoll sind. Ich möchte aufzeigen, welche Möglichkeiten es hier gibt da verschiedene Anwendungen ver-

1.2 Begriffsabgrenzung

schiedene Ansätze benötigen. So ist ein energieverbrauch-optimierendes Routing eher an dem 99%-Quartil interessiert, während ein stau-minimierendes Routing eher die Durchschnittszeit betrachten wird. Ich habe es mir zum Ziel gesetzt, die nötigen Anpassungen meines Verfahrens zu beschreiben, aber nicht für jedes Anwendungsgebiet die passende Speziallösung zu finden. Exemplarisch werde ich dafür die genannten Routing-Verfahren beleuchten.

1.2 Begriffsabgrenzung

In der Fachliteratur ([Chawla and Roughgarden, 2006]) ist der Begriff „stochastic routing“ oft etwas anders definiert als in meiner Arbeit. In vielen Arbeiten ist darunter das Routing zu verstehen, welches für jede Strecke in einem Netz eine Wahrscheinlichkeit verknüpft, mit der diese überhaupt zwei Knoten verbindet. Die Fragestellung ähnelt meiner zwar dahin gehend, dass ebenfalls eine Wahrscheinlichkeit maximiert werden soll, sie unterscheidet sich aber grundsätzlich in der mathematischen Beschreibung und Lösung.

1.3 Wissensstand

Es gibt bereits Arbeiten, die sich mit diesem oder einem ähnlichen Thema beschäftigen. Ein Paper aus 2011 ([Berger et al., 2011]) nutzt stochastisches Routing mit diskreten Verteilungen, um Verspätungen der Deutschen Bahn vorherzusagen. Die dort verwendete Methode kommt meiner am nächsten.

Ich bin der Meinung, dass das verwendete Modell in vielen Arbeiten durch die Wahl der Wahrscheinlichkeitsverteilung einige Probleme aufweisen, die ich mit dem von mir vorgeschlagenen Algorithmus zu lösen versuche. Diese Probleme entstehen durch die Wahl einer kontinuierlichen Verteilung, was ich in dem Kapitel *Lösungsweg* genauer beleuchten werde.

1 Einleitung

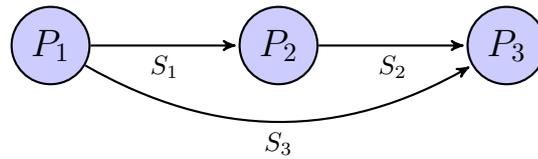


Abbildung 1.1: Beispielgraph

1.4 Ausgangssituation

In diesem Abschnitt möchte ich meine Fragestellung erläutern und skizzieren, wie eine Lösung aussehen kann. Meine Fragestellung lautet:

Wie kann man den Weg durch einen Graphen finden, welcher wahrscheinlich die niedrigsten Kosten hat?

Dies möchte ich an einem Beispiel(1.1) verdeutlichen:

Um von dem Knoten P_1 zum Ziel P_3 zu kommen, können wir entweder direkt (über die Kante S_3) oder über den Knoten P_2 und damit über die Kanten S_1 und S_2 gelangen. Ich habe mich damit beschäftigt, welche Beschreibungen von Kanten dafür geeignet sind und wie es sich überprüfen lässt, ob S_3 oder S_1 verknüpft mit S_2 mit größerer Wahrscheinlichkeit ans Ziel führt oder durchschnittlich kürzer ist.

S_1 , S_2 und S_3 sind dabei als Zufallsvariable definiert. Im Gegensatz zu dem bekannten Routing, bei dem Kanten einen festen Wert (zum Beispiel benötigte Zeit oder als Entfernung) zugewiesen bekommen, können beim stochastischen Routing die Kanten immer andere Werte annehmen. Diese Werte, die einer Kante zugewiesen werden können, sind nicht gleichmäßig verteilt. Unterschiedlichen Werten werden unterschiedliche Wahrscheinlichkeiten zugewiesen.

1.5 Problembeschreibung

Formell stellt sich das Problem wie folgt dar:

1.5 Problembeschreibung

Seien zwei Punkte $P1$ und $P2$ gegeben, die durch die Strecke S verbunden sind. f_S ist nun die Funktion, die jeder Möglichen Distanz, unabhängig von dem gewählten Maß, genau einen Wert zuordnet. Dabei bezeichnet D den Definitionsbereich, üblicherweise \mathbb{R} .

$$f_S : D \rightarrow \mathbb{R}$$

Hierbei ergeben sich die folgenden Einschränkungen: Da es sich um eine stochastische Funktion handelt, muss gelten

$$\int_D f(x) dx = 1$$

Betrachten wir nun zwei Strecken $S1$ und $S2$. Seien f und g die zugehörigen Funktionen. Wir suchen nun die Strecke $\overline{S1S2}$. Diese Strecke ist die Verknüpfung zweier Funktionen. Um eine Vergleichbarkeit und Stabilität zu garantieren, muss diese Funktion selbstähnlich sein, d.h. f verknüpft mit g muss wieder eine Funktion mit den selben Einschränkungen ergeben. Deshalb muss die Summe der Werte der entstanden Funktion ebenfalls 1 ergeben.

$$\int_D (f * g) = \int_D f \cdot \int_D g = 1$$

Bei meinen Nachforschungen bin ich auf zwei weitere Eigenschaften gestoßen, die für meinen speziellen Anwendungsfall erforderlich sind. Dies sind erstens die Möglichkeit, die Definitionsmenge einzuschränken. Außerdem ist es notwendig, einen endlichen Erwartungswert und endliche Varianz zu haben, da Zeitabschätzungen für Funktionen mit unendlich großem Erwartungswert nicht sinnvoll sind.

1 Einleitung

1.6 Ziel meiner Arbeit

Ich habe es mir zum Ziel gesetzt, eine geeignete Repräsentation einer Wahrscheinlichkeitsverteilung zu finden, die alle Einschränkungen erfüllt und sich für den praktischen Einsatz in einer Navigationssoftware eignet. Außerdem möchte ich ihre mathematische Korrektheit und ihre Allgemeingültigkeit überprüfen. Darüber hinaus möchte ich eine einfache Implementierung entwerfen und meine Ergebnisse evaluieren.

2 Lösungsweg

Im Folgenden möchte ich die Methoden beschreiben, die ich genutzt habe, um die mir gestellte Aufgabe zu lösen.

Als erstes habe ich damit begonnen, die Domäne meines Problems zu beschreiben. Dadurch konnte ich die Kriterien ermitteln, die eine Lösung aufweisen muss, um nach meiner Definition korrekt zu sein. Auch möchte ich sicherstellen, dass ich alle Lösungen betrachtet habe. Daher beginne ich damit, zu beschreiben, was ich unter *Routing* überhaupt verstehe.

2.1 Routing

Routing als Oberbegriff sei wie folgt definiert:

Sei G ein Graph mit $G = (V, E)$. Jetzt ordnen wir jeder Kante E zwischen zwei Knoten V_i und V_j eine Wahrscheinlichkeitsverteilung $W(V_i, V_j)$ zu. Des Weiteren sei ein Startknoten V_0 und ein Zielknoten V_n gegeben. Wir suchen jetzt die Verknüpfungen der Kanten, die V_0 mit V_n verbindet und die minimale Verteilung $W(V_0, \dots, V_n)$ besitzt. Diese definieren wir als

$$W(V_0, \dots, V_n) = W(V_0, V_1) + \dots + W(V_{n-1}, V_n)$$

Gesucht sei nun eine geeignete Funktion W , die sich für eine algorithmische Umsetzung eignet. Dafür werde ich in dem folgenden Kapitel alle möglichen Verteilungen aufzeigen und vergleichen.

2 Lösungsweg

2.1.1 Stochastische Kostenfunktion

Im gängigen Routing ist die Kostenfunktion einer Kante entweder die skalare Distanz zwischen den beiden Knoten oder die benötigte Zeit für die Strecke. Da ich mich mit stochastischem Routing beschäftige, muss die Kostenfunktion aus vielen Werten mit einer *Gewichtung* bestehen. Je wahrscheinlicher ein bestimmter Wert ist, desto größer ist seine Gewichtung. Über die Gewichtungen wissen wir, dass sie zusammen immer 100 Prozent ergeben müssen, da wir damit alle Fälle abgedeckt haben. So eine Zuordnung von einer Wahrscheinlichkeit zu einem Wert ist als *Wahrscheinlichkeitsverteilung* bekannt. Damit wissen wir, dass wir eine solche Verteilung als Kostenfunktion definieren müssen.

In den folgenden Abschnitten möchte ich die verschiedenen Arten von Verteilungen untersuchen.

2.2 Verteilungsfunktion und Wahrscheinlichkeitsfunktion

Wahrscheinlichkeitsfunktionen ordnen bestimmten Werten eine gewisse Wahrscheinlichkeit zu. Im Folgenden werde ich untersuchen, wie sich verschiedene Arten von Funktionen verhalten, wenn sie im Routing genutzt werden, um eine Strecke zu beschreiben. Insbesondere Kombination dieser ist wichtig, da dieser Vorgang der elementare ist, um mehrere Strecken kombinieren zu können.

Jede Verteilung hat zwei Darstellungsformen, von denen sich eine aus der anderen gewinnen lässt: die (kumulative) Verteilungsfunktion, die die aufsummierten Wahrscheinlichkeiten darstellt, sowie die Wahrscheinlichkeitsfunktion, aus der man die einzelnen Wahrscheinlichkeiten direkt ermitteln kann.

Um eine Verteilung darzustellen, gibt es zwei Möglichkeiten: Entweder ordnet man tabellenartig jedem Wert eine Wahrscheinlichkeit zu (diskrete Verteilung), oder man sucht eine geeignete Funktion (kontinuierliche Verteilung), die diese Zuordnung bestimmt. Eine diskrete Wahrscheinlichkeitsfunktion und die zugehörige Verteilungsfunktion sind in

2.2 Verteilungsfunktion und Wahrscheinlichkeitsfunktion

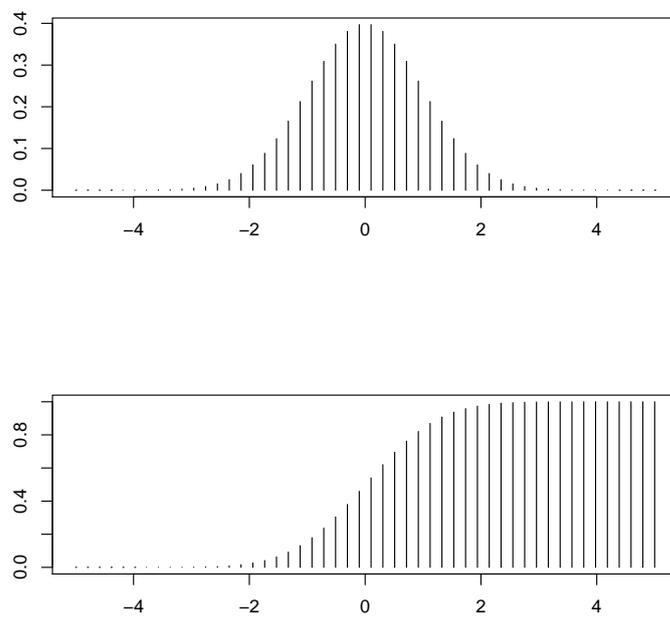


Abbildung 2.1: Beispiel einer diskreten Wahrscheinlichkeits- und der zugehörigen Verteilungsfunktion

2 Lösungsweg

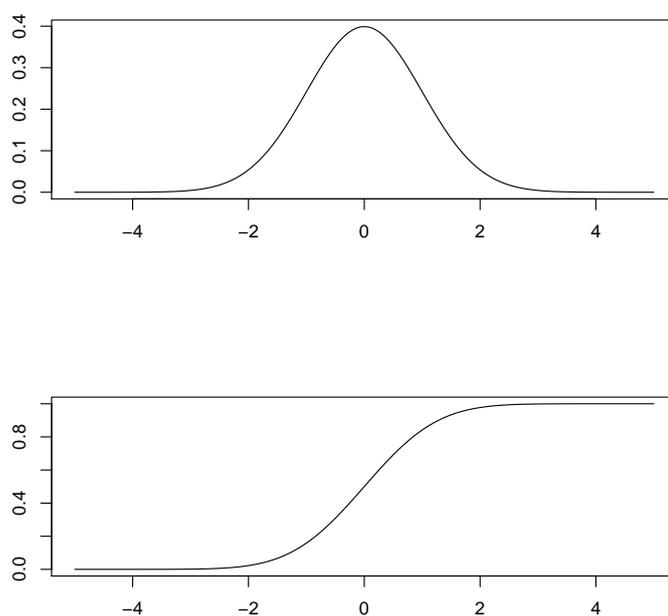


Abbildung 2.2: Beispiel einer kontinuierlichen Wahrscheinlichkeits- und der zugehörigen Verteilungsfunktion

2.3 Anforderungen an eine stochastische Kostenfunktion

Abbildung 2.1 dargestellt. Es gibt natürlich auch gemischte Ansätze, die ich hier aber nicht betrachten werde.

Ich möchte in den nächsten Abschnitten meiner Arbeit die Vor- und Nachteile beider Arten beleuchten und diese in den Kontext meiner Anforderungen stellen.

2.3 Anforderungen an eine stochastische Kostenfunktion

Um für Routing genutzt werden zu können, müssen kontinuierliche Funktionen im Kontext ihrer Verknüpfung betrachtet werden. Eine Verknüpfung zweier Gewichtungsfunktionen sollte nicht anders beschrieben werden müssen als die beiden Ausgangsfunktionen. Dies bedeutet, dass für eine Konkatenation

$$W_i + W_j = W'_k$$

gelten muss, dass W' die gleiche Beschreibungskomplexität wie W aufweisen muss. Sollte dies nicht der Fall sein, würde die Beschreibungskomplexität abhängig sein von der Problemgröße und damit wäre eine praktische Umsetzung schwer, da Speicherbedarf und Laufzeit mit der Größe des Problems stark wachsen würden. Außerdem wären Vergleichskriterien für W unterschiedlicher Komplexität wie Erwartungswert oder Varianz nicht allgemeingültig bestimmbar sondern (wenn überhaupt) nur analytisch berechenbar. Daher stelle ich als Anforderung auf, dass eine Kostenfunktion im stochastischen Routing selbstständig bezüglich der Kombination sein muss.

Um bestimmen zu können, welche Funktionen sich ohne Änderung der Beschreibungskomplexität kombinieren lassen, untersuchen wir zunächst die Definition der Kombination.

2 Lösungsweg

2.3.1 Faltung von Verteilungen

Die Konkatenation von Wahrscheinlichkeitsverteilungen ist mathematisch eine Faltung. Sie stellt also nicht die einfach Summe aller Werte dar, was ja bei der Kombination von Skalaren beim Routing gemacht wird, sondern eine Multiplikation im Frequenzbereich. Die Faltung zweier Funktionen f und g ist wie folgt definiert:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau$$

Eine alternative Definition wäre die über das Produkt im Frequenzbereich. Sei \mathcal{F} die Fouriertransformation und \mathcal{F}^{-1} die inverse Fouriertransformation:

$$f * g = \mathcal{F}\{\mathcal{F}^{-1}\{f\} \cdot \mathcal{F}^{-1}\{g\}\}$$

Wir werden später sehen, dass diese Definition bei der Berechnung Vorteile bringen kann. Gesucht seien jetzt alle Verteilungen, für die gilt

$$W_i + W_j \text{ ist selbstähnlich zu } W_{i+j}$$

Im nächsten Kapitel möchte ich die verschiedenen Verteilungen, die diese Anforderung erfüllen, untersuchen.

2.4 Arten von Verteilungen

2.4.1 Alpha-stabile Verteilungen

Zum Beginn meiner Arbeit habe ich versucht, eine geeignete kontinuierliche Verteilung zu finden, da diese nicht nur mathematisch kürzer zu beschreiben sind, sondern auch beliebig präzise. Jede diskrete Verteilung ist als Abtastung einer idealen Verteilungsfunktion interpretierbar und kann damit nur genähert werden, während eine kontinuierliche

Verteilung zumindest in der Theorie exakt sein kann.

Man könnte annehmen, dass jede Funktion, deren Integral 1 beträgt, in Frage kommen könnte doch nicht jede Verteilungsfunktion ist geeignet. Paul Lévy hat bereits in den 1920er Jahren zeigen können [Nolan, 2007], dass es nur drei als Formel beschreibbare Verteilungen gibt, die *alpha-stabil* sind (Es gibt noch weitere, nicht als Formel darstellbare Funktionen, die ich hier aber nicht näher betrachten werde). Eine alpha-stabile Verteilung (oder Lévy-stabile Verteilung) ist insofern stabil, dass eine Linearkombination aus zwei Verteilungen sich wieder als eine darstellen lässt. Formell lässt sich das wie folgt beschreiben

$$aX_1 + bX_2 \stackrel{d}{=} cX + d$$

Damit erfüllt diese Klasse von Verteilungen meine Einschränkung einer Verteilungsfunktion. Nun werde ich auf die Ergebnisse von Paul Lévy eingehen und die verschiedenen Verteilungen vorstellen. Diese sind die

- Gaußverteilung
- Cauchy-Verteilung
- Lévy-Verteilung

Gaußverteilung

Die bekannteste alpha-stabilen Verteilungen ist die Gaußverteilung. Diese ist definiert durch

$$X \sim N(\mu, \sigma^2)$$
$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

Die Faltung von zwei Gaußverteilungen wird wie folgt beschrieben:

2 Lösungsweg

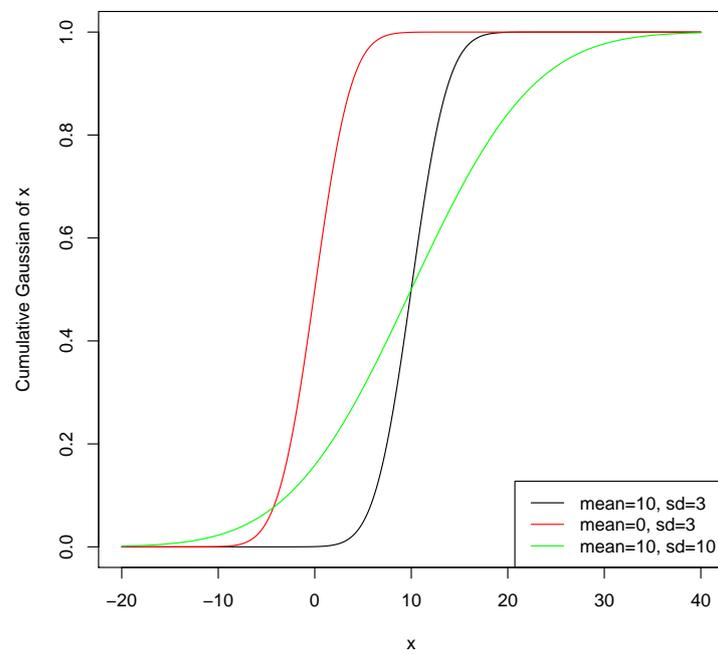
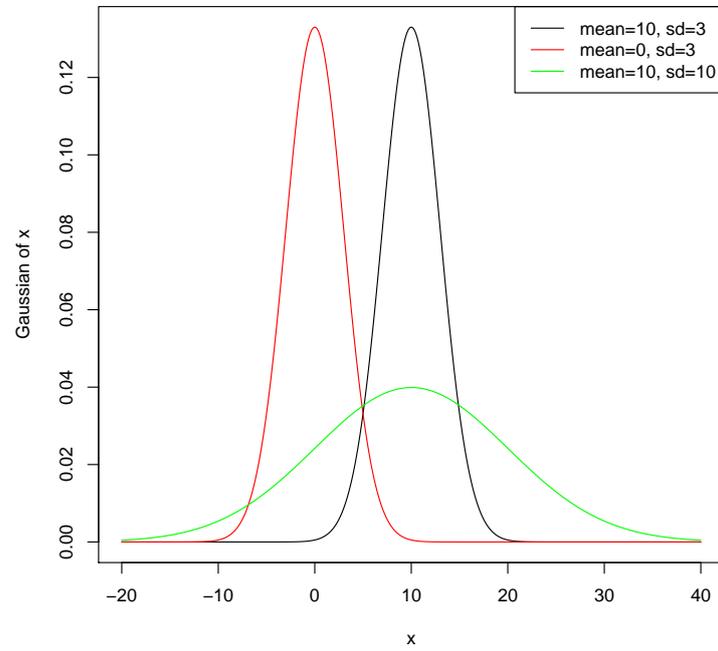


Abbildung 2.3: Drei Normalverteilungen

$$\begin{aligned}X &\sim N(\mu_X, \sigma_X^2) \\Y &\sim N(\mu_Y, \sigma_Y^2) \\Z &= X + Y \\ \Rightarrow Z &\sim N(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2)\end{aligned}$$

Damit ist, wie bereits festgestellt, die Faltung beider Funktionen wieder selbständig zu den Ausgangsfunktionen und lässt sich mit der gleichen Anzahl an Variablen komplett beschreiben.

Um die Normalverteilung als Gewichtsfunktion zu nutzen, können wir als vergleichende Metriken den Erwartungswert μ , den Median μ , die Varianz σ^2 oder eine beliebige Kombination nehmen.

Die Gaußverteilung hat allerdings zwei Eigenschaften, die Probleme beim Routen verursachen können: Sie hat keine Definitionsgrenzen. Da wir nicht in jedem Fall mit negativen Werten für Akkuladestände oder Zeit rechnen wollen, ist diese Verteilung nicht optimal.

Cauchy-Verteilung

Die Cauchy-Verteilung, definiert durch

$$f(x) = \frac{1}{\pi} \cdot \frac{\gamma}{\gamma^2 + (x - \delta)^2}$$

Die Funktion eignet sich allerdings nicht für die Lösung des Problems, da die Cauchy-Verteilung keinen endlichen Erwartungswert und keine Varianz hat. Damit sind keine sinnvollen Vergleichsmetriken gegeben. Außerdem besitzt diese Verteilung ebenfalls wie die Gaußverteilung keine Definitionsgrenzen.

2 Lösungsweg

Lévy-Verteilung

Das lässt uns mit der letzten Verteilung - der Lévy-Verteilung. Diese wird beschrieben durch

$$f(x) = \sqrt{\frac{\gamma}{2\pi}} \frac{1}{(x - \delta)^{3/2}} \exp\left(-\frac{\gamma}{2(x - \delta)}\right)$$

bzw durch die Verteilungsfunktion

$$F(x) = \operatorname{erfc}\left(\sqrt{\frac{\gamma}{2(x - \delta)}}\right)$$

Die Faltung zweier Lévy-Verteilungen ist gegeben durch:

$$X \sim \operatorname{Levy}(\delta_X, \gamma_X)$$

$$Y \sim \operatorname{Levy}(\delta_Y, \gamma_Y)$$

$$Z = X + Y$$

$$Z \sim \operatorname{Levy}(\delta_X + \delta_Y, (\sqrt{\gamma_X} + \sqrt{\gamma_Y})^2)$$

Die Lévy-Verteilung hat als einzige der drei betrachteten Verteilungen die Eigenschaft, begrenzt zu sein. Sie ist definiert auf dem Intervall

$$x \in [\delta, \infty)$$

Im frühen Stadium meiner Arbeit habe ich eine Implementierung auf Basis dieser Verteilung für die Lösung gehalten. Der Schnittpunkt zweier Lévy-Verteilungen ist eindeutig für zwei Verteilungen mit $\gamma_1 \neq \gamma_2$. Der Umstand, dass die *erfc*-Funktion stetig ist, können wir den Schnittpunkt so berechnen:

2.4 Arten von Verteilungen

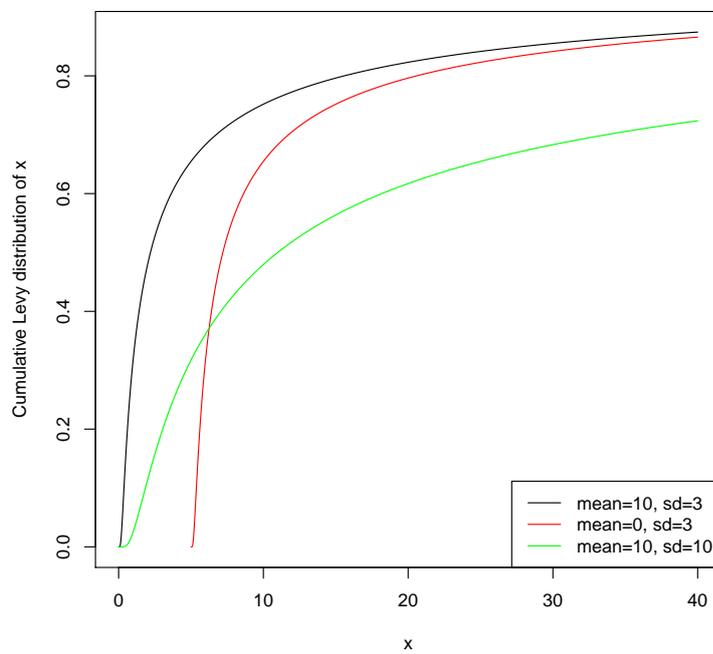
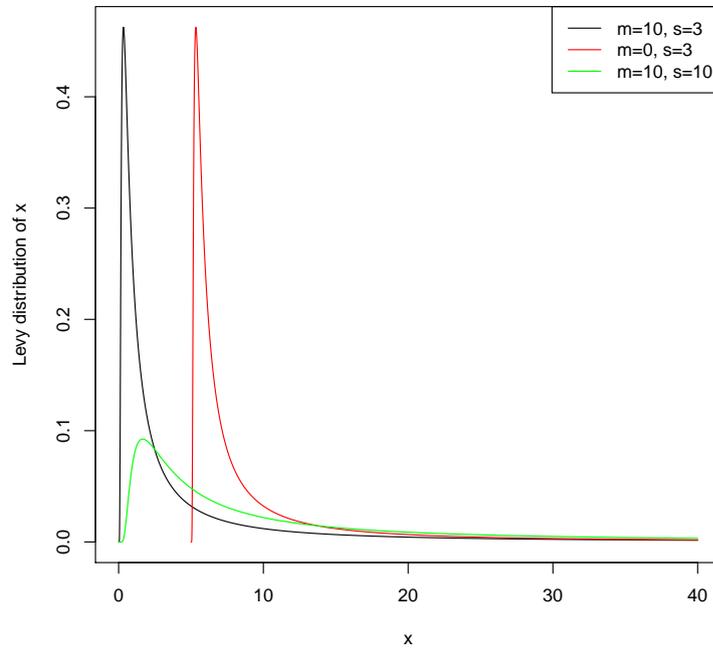


Abbildung 2.4: Drei Lévy-Verteilungen

2 Lösungsweg

$$F_1(x) = F_2(x)$$
$$x = \frac{\delta_1\gamma_2 - \delta_2\gamma_1}{\gamma_2 - \gamma_1}$$

Somit können wir für zwei Verteilungen einfach und schnell berechnen, ab welchem Wert eine der anderen vorgezogen werden sollte. Sollte es keinen Schnittpunkt geben, müssen wir die weiteren Wege, die die zugeordnete Kante nutzen, nicht berechnen, da diese immer schlechtere Werte liefert. Dies sind alles ideale Eigenschaften für den Einsatz im stochastischem Routing.

Die Lévy-Verteilung hat jedoch einen entscheidenden Nachteil: Da sie ebenfalls keinen endlichen Erwartungswert und keine endliche Varianz besitzt, kann für eine ermittelte Route keine mathematisch korrekte durchschnittliche Dauer und kein korrekter Akkuverbrauch errechnet werden. Hier könnte eine Approximation helfen, bei der die Wahrscheinlichkeiten unter einem gewissen Schwellwert mit 0 gleichgesetzt werden. Diese Methode ist jedoch weder exakt, noch lässt sich mit der entstehenden, abgeschnitten Verteilung nicht weiter rechnen, ohne die Beschreibungskomplexität zu erhöhen.

Diese Ergebnisse haben mich zu dem Schluss kommen lassen, dass eine kontinuierliche Verteilungsfunktion nicht die von mir gestellten Anforderungen an ein Verfahren für das stochastische Routing erfüllt.

2.5 Diskrete Verteilungen

Wie wir im letzten Abschnitt gesehen haben, eignen sich die kontinuierlichen Verteilungen nicht optimal, da jede in Frage kommende Verteilungsfunktion mindestens einen Nachteil aufweist. Als letzte verbliebende Option möchte ich jetzt die diskrete Verteilung auf ihre Tauglichkeit überprüfen.

Eine Beispiel-Verteilung könnte so aussehen:

Akkuverbrauch	0	0.25	0.5	0.75	1
Wahrscheinlichkeit	0	0.5	0.5	0	0

Wenn wir jetzt diese Strecke doppelt fahren würden, erhielten wir eine Verteilung, die wie folgt aussehen würde:

Akkuverbrauch	0	0.25	0.5	0.75	1
Wahrscheinlichkeit	0	0	0.25	0.5	0.25

2.5.1 Definition

Im Gegensatz zu den kontinuierlichen Verteilungen wird die diskrete Verteilung nur von einem Parameter beschrieben: Der Anzahl der Werte.

Die Verknüpfung zweier Verteilungen wird wieder durch die Faltung errechnet. Diese ist für diskrete Verteilungen definiert als:

Wir haben zwei diskrete Funktionen f und g , die von dem Definitionsbereich D nach \mathbb{R} mit einem Wertebereich

$$\{-M, -M + 1, \dots, M - 1, M\}$$

abbilden. Die Faltung dieser beiden Funktionen ist nun definiert als

$$(f * g)[n] \stackrel{\text{def}}{=} \sum_{k=-M}^M f(k)g(n - k) \quad (2.1)$$

2.5.2 Beschreibungskomplexität

Aus der Definition 2.1 geht hervor, dass wir für die Faltung von n Werten ($-M$ bis M) genau $2n - 1$ Ergebnisse erhalten. Das verstößt wieder gegen die Forderung, selbstähnlich

2 Lösungsweg

zu sein, da der beschreibende Parameter der Verteilung, nämlich die Anzahl der Werte, mit jeder Faltung steigen würde. Um dieses Problem zu lösen, habe ich die entstehenden Werte reduziert. Diese Reduktion besteht darin, jeden zweiten Wert je zur Hälfte auf die umgebenden Werte zu verteilen. Dies lässt sich formell so beschreiben:

Seien f, g zwei diskrete Verteilungen und $f * g$ die nach 2.1 definierte Faltung. Die Anzahl der Werte einer diskrete Verteilung sei mit $|f|$ beschrieben. So ist eine reduzierte Funktion $(f * g)_{red}$, die alle Eigenschaften der Verteilung erhält, so definiert:

$$\begin{aligned} |f| &= |g| = n \\ |f * g| &= 2n - 1 \\ (f * g)_{red}[x] &= \begin{cases} (f * g)[0] + 0.5 \cdot (f * g)[1] & \text{für } x = 0 \\ (f * g)[2n - 1] + 0.5 \cdot (f * g)[2n - 2] & \text{für } x = n \\ (f * g)[2x - 1] + 0.5 \cdot ((f * g)[2x - 2] + (f * g)[2x]) & \text{sonst} \end{cases} \\ |(f * g)_{red}| &= n \end{aligned}$$

Damit erhalten wir für jede Verknüpfung von zwei diskreten Verteilungen mit jeweils n Werten wieder genau n Werte und die Selbstähnlichkeit bleibt erhalten.

2.5.3 Beispiel

Um zu sinnvollen Ergebnissen zu kommen, sind natürlich weitaus mehr Werte notwendig, als in den beiden Tabellen aus der Einleitung. Um eine komplexere Rechnung darzustellen, habe ich 100 Werte aus einer Normalverteilung abgetastet und diese diskret gefaltet. Das Ergebnis sieht wie folgt aus:

In der Abbildung 2.5 werden drei Funktionen dargestellt: f, g und eine dritte, welche die Faltung der beiden Funktionen darstellt ($(f * g)$). Hier erkennt man auch die typischen Eigenschaften der Faltung: Die Varianz und der Durchschnittswert nehmen zu.

2.5 Diskrete Verteilungen

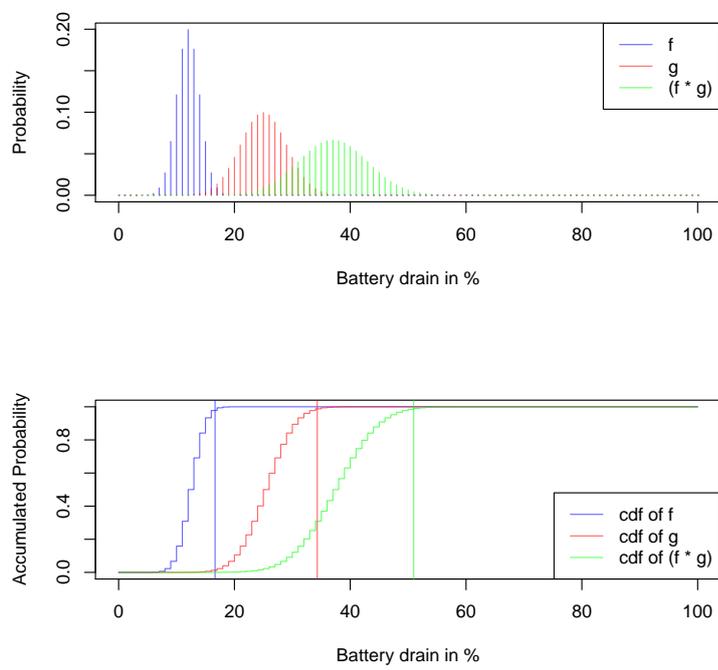


Abbildung 2.5: Beispiel zweier diskreter Verteilungen und ihrer Faltung

2 Lösungsweg

In der unteren Abbildung sind die 99%-Quantile der jeweiligen Verteilungen eingezeichnet. Mit diesen Werten können wir eine Reichweitenanalyse durchführen. Gleichzeitig können wir die Durchschnittswerte betrachten (nicht eingezeichnet), um die Strecke mit dem Wahrscheinlich geringsten Verbrauch zu finden.

3 Ergebnisse

In diesem Kapitel möchte ich die in dem vorigen Kapitel herausgearbeiteten Verteilungen vergleichen. Ich werde zu dem Ergebnis kommen, dass die diskrete Verteilung als einziges meine Anforderungen erfüllt und die wenigsten Nachteile aufweist.

3.1 Vergleichskriterien

Um Verteilungen zu vergleichen, müssen wir zwei Ebenen betrachten. Zuerst ist es wichtig, die mathematischen Eigenschaften wie die Existenz eines Erwartungswertes zu betrachten. Diese Eigenschaften sind unabhängig von dem tatsächlichen Routing und universell. Darüber hinaus existiert eine spezifische Komponente, nämlich die Form der Verteilung.

3.1.1 Vergleich der Form

Auf die spezifische Form der Verteilungen möchte ich hier nur kurz eingehen, da ich über keine Messwerte verfüge, um mein Thesen zu unterstützen und nur auf nicht für diesen Zweck angefertigte Messreihen anderer Arbeit zurückgreifen kann. Wie in Grafik 3.1 zu sehen, hat jede kontinuierliche Verteilung eine charakteristische Form. Die Cauchy-Verteilung ist hier weggelassen worden, da sie sich visuell nicht sehr von der Gaußverteilung unterscheidet.

Der auffälligste Unterschied der beiden kontinuierlichen Verteilungen in der Grafik ist das Vorhandensein einer Symmetrie. Mit der Lévy-Verteilung lassen sich nur beschränkte, asymmetrische Verteilungen gut approximieren, während die Gaußverteilung nur für symmetrische, unendliche Verteilungen geeignet ist. Multimodale Verteilungen lassen

3 Ergebnisse

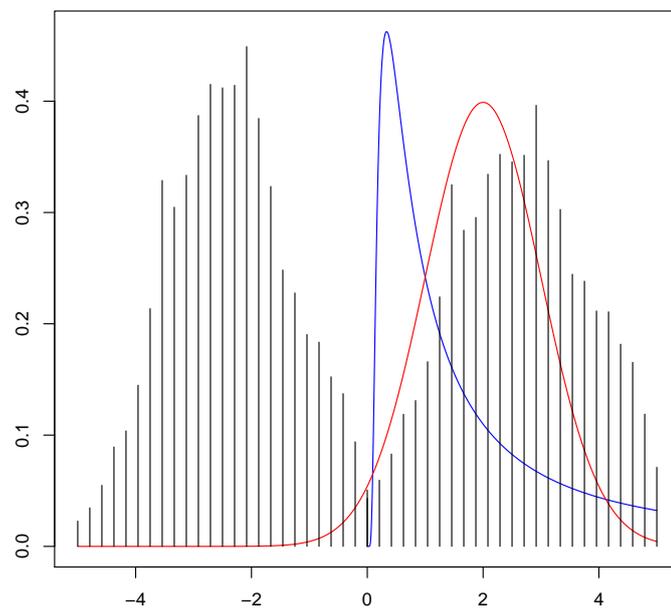


Abbildung 3.1: Eine Gaußverteilung (Rot), eine Lévy-Verteilung (Blau) und eine diskrete Verteilung (Schwarz)

sich gar nicht abbilden, was mit einer diskreten Verteilung hingegen problemlos möglich ist (im Bild dargestellt).

3.1.2 Vergleich der mathematischen Eigenschaften

In der Domäne des Routing gibt es zwei Eigenschaften von Verteilungen, ohne die Routing nicht so einfach möglich ist. Diese sind

- Existenz eines Erwartungswerts und einer Varianz
- Beschränktheit

Der erste Punkt ist notwendig, damit wir die Verteilungen vergleichen können. Da weder die Lévy-Verteilung noch die Cauchy-Verteilung einen Erwartungswert besitzen, eignen sich diese beiden nicht für das stochastische Routing.

Egal, ob wir den geringsten Akkuverbrauch oder die schnellste Strecke suchen - weder ein Akkuverbrauch über 100 Prozent noch eine negative Dauer sinnvoll. Von den kontinuierlichen Verteilungen ist nur die Lévy-Verteilung beschränkt. Damit bleibt als letztes die diskrete Verteilung übrig, die alle Kriterien erfüllt.

4 Lösung

In diesem Kapitel möchte ich je eine vollständige Spezifikation für zwei Anwendungsfälle vorstellen. Danach werde ich eine einfache Implementierung vorstellen, diese erweitern und schließlich beschreiben, wie man eine optimierte Variante entwickeln könnte.

4.1 Spezifikation

4.1.1 Reichweitenoptimierung

Wenn wir nur den Energieverbrauch einer endlichen Ressource wie eines Akkus betrachten, reicht es, die Wahrscheinlichkeiten aller Akkuänderungen von -100 Prozent bis 100 Prozent zu betrachten (vollständiges Aufladen bis zum vollständigen Entladen). Eine Speicherung dieser Grenzwerte ist nicht notwendig, da sich die Ergebnisse niemals außerhalb der Grenzen befinden können, ohne die Rechnung ungültig zu machen (möglicher Verbrauch von über 100 Prozent).

4.1.2 Fahrzeitminimierung

Würden wir das gleiche Modell jedoch für die Dauer einer Strecke benutzen, hätten wir ein Problem: wir müssten die Grenzen vorher wissen. Diese könnten unter Umständen so weit auseinander liegen, dass die meisten Einträge der Verteilung aus Nullen bestehen. Damit hätten wir viel redundante Information. Dies können wir optimieren, indem wir für jede Verteilung den ersten und den letzten Wert, der nicht Null ist, mitspeichern. Wenn wir nun zwei Verteilungen miteinander Falten, sind die neuen Grenzen die Addition der Grenzen der Ausgangsverteilungen. Mit diesem Modell haben wir außerdem ein weiteres Vergleichskriterium: die Spanne, über die die Verteilung definiert ist. Dieses Modell sollte unbedingt bei einer praktischen Umsetzung genutzt werden.

4 Lösung

4.2 Algorithmische Komplexität

Da die Umsetzbarkeit und die Geschwindigkeit meiner Lösung vor allem von der diskreten Faltung abhängt, werde ich eine einfache Lösung präsentieren und zeigen, wie man die Laufzeit verbessern könnte.

4.2.1 Summenformel

Der naive Ansatz besteht in der direkten Umsetzung der Summenformel der diskreten Faltung.

Listing 4.1: Implementierung der diskreten Faltung

```
1 // f and g are double arrays of equal length
2 int right_bound = length(f) - 1;
3 int left_bound = -right_bound;
4
5 int position, n;
6
7 for (position = left_bound; position <= right_bound; position++) {
8
9     int from = position <= 0 ? 0      : position;
10    int to =   position >= 0 ? right_bound : right_bound + position;
11
12    double sum = 0;
13
14    for (n = from; n <= to; n++) {
15        sum += f[n] * g[n - position];
16    }
17
18    // sum is now (f * g)[n]
19 }
```

Um die Laufzeitkomplexität dieser Lösung zu ermitteln, betrachten wir zunächst die äußere Schleife. Wenn wir zwei Arrays f und g von der Länge n haben, ist der Abstand zwischen $left_bound$ und $right_bound$ genau $2n - 1$. In der inneren Schleife variiert die Laufweite, überschreitet aber nie n . Das wird deutlich, wenn man eine Fallunterschei-

ung für die Variable *position* macht. Für Werte kleiner als 0 ist der maximale Abstand *right_bound*, für Werte größer als 0 beträgt er ebenfalls *right_bound*. Da *right_bound* als $n - 1$ definiert wurde, ergibt sich eine Laufzeitkomplexität von

$$(2n - 1)(n - 1) = 2n^2 - 3n + 1$$
$$\Rightarrow \lim_{n \rightarrow \infty} 2n^2 - 3n + 1 \leq \lim_{n \rightarrow \infty} a \cdot n^2$$

Wir sehen also, dass dieser Algorithmus für die diskrete Faltung in $\mathcal{O}(n^2)$ liegt. Dies ist kein gutes Ergebnis für beispielsweise tragbare Geräte, da dort der Arbeitsspeicher und die Rechengeschwindigkeit begrenzt sind. Trotzdem zeigt dieses Beispiel, dass meine Lösung relativ einfach umzusetzen ist.

4.2.2 Mit Nutzung der FFT

Es gibt eine einfache Verbesserung zum naiven Ansatz: Wenn wir die Eingangswerte zyklisch fouriertransformieren, ist die Faltung dieser lediglich die einfache Multiplikation. Die Zeitkomplexität der diskreten Fouriertransformation ist nicht besser als die des nativen Ansatzes, aber unter Nutzung der schnellen Fouriertransformation können wir eine Laufzeit von $\mathcal{O}(n \log(n))$ erreichen. Der Reduktionsschritt entfällt hier allerdings auch nicht.

5 Schlussfolgerung

In diesem Abschnitt möchte ich eine mögliche Evaluation beschreiben; vorschlagen, wie man ihn in der Praxis einsetzen könnte und die Vor- und Nachteile meines Algorithmus untersuchen.

5.1 Evaluation

Eine Evaluation meiner Ergebnisse war im Rahmen meiner Arbeit nicht in vollem Umfang durchführbar, da es nötig gewesen wäre, ein Navigationssystem mit dem Modell zu implementieren und diverse Testfahrten durchzuführen.

Es gibt zwei unterschiedliche Arten von Evaluationen: einmal können die Eigenschaften vom stochastischen Routing selbst mit denen normalen Routings verglichen werden. Die andere Möglichkeit, die für meine Arbeit relevanter ist, ist der Vergleich verschiedener Wahrscheinlichkeitsfunktionen.

Wie ich ja herausgearbeitet habe, kommen dafür nur drei kontinuierliche Funktionen in Frage. Diese haben alle unterschiedliche Eigenschaften, was einen Vergleich schwierig macht.

5.1.1 Versuchsaufbau für eine mögliche Evaluation

Für eine aussagekräftige Evaluation ist es notwendig, die Performanz der einzelnen Algorithmen zu vergleichen. Es muss also für jede Wahrscheinlichkeitsfunktion (Gauß-Verteilung, Cauchy-Verteilung, Lévy-Verteilung) das Modell gefunden werden, was der Realität am ehesten entspricht. Diese Modelle könnten dann in eine Navigationssoftware, wie dem Greennav-Projekt, implementiert werden. Damit wäre es möglich, Tests durch-

5 Schlussfolgerung

zuführen und die Ergebnisse der einzelnen Modelle miteinander zu vergleichen.

Ich bin der Meinung, dass sich von den drei kontinuierlichen Verteilungsfunktionen die Lévy-Verteilung am besten eignen wird. Viele Daten aus anderen Arbeiten weisen eine Form auf, die nicht symmetrisch ist. Diese Werte lassen sich besser durch eine Lévy-Verteilung approximieren als durch eine Gaußverteilung. Besonders bei kurzen Strecken fällt außerdem auf, dass ein großer Teil der Gaußverteilung im negativen Bereich ist. Dies ist bei der Lévy-Verteilung nicht möglich, da diese begrenzt ist.

5.2 Vorteile

5.2.1 Modellbildung

Ich möchte hier nicht näher darauf eingehen, wie eine mögliche Funktion ermittelt wird, die aus den Eigenschaften einer Strecke eine Wahrscheinlichkeitsverteilung errechnet, da dies nicht zu meiner Arbeit gehört. Eine diskrete Verteilung ermöglicht es jedoch, Voraussagen für eine Strecke zu machen, indem die Werte mehrfach gemessen werden. Diese *Häufigkeitsverteilung* nähert sich, bei ausreichend großer Fallzahl, der wirklichen Verteilung an (Zeitliche Invarianzen werden hier nicht betrachtet). Diese Messwerte könnten also direkt für die Modellbildung herangezogen werden, was einen enormen Vorteil bietet.

5.2.2 Adaption

Wie bereits besprochen, können multimodale Verteilung ebenso wie symmetrische oder asymmetrische dargestellt und miteinander kombiniert werden. Eine Approximation ist nicht notwendig, wie es bei kontinuierlichen Verteilungen der Fall ist. Dieses Modell ermöglicht es ebenfalls, die Vorhersage über eine Strecke an die Wirklichkeit anzupassen, wenn man gefahrene Strecken mit abnehmender Gewichtung dem Modell hinzufügt. Damit könnte ein Navigationsgerät lernfähig und mit jeder gefahrenen Strecke genauer werden. Dies stellt einen enormen Vorteil dar, der so beim klassischen Routing nicht genutzt werden kann.

5.2.3 Echtzeitfähigkeit

Das Modell benötigt keinen Vorverarbeitungsschritt und ist, durch die gering höhere Laufzeit, die durch die Verknüpfung der Kantengewichte in $\mathcal{O}(n \log(n))$ entsteht, potentiell Echtzeitfähig. Auf moderner Hardware habe ich einen Durchsatz von vielen tausend Verknüpfungen pro Sekunde erreicht. Die genaue Geschwindigkeit hängt jedoch von zu viele Faktoren ab, um eine allgemein gültige Vorhersage zu treffen.

5.3 Nachteile

5.3.1 Rundungsfehler

Da die Werte der Verteilungen im Computer mit Gleitkommazahlen gespeichert werden, tritt natürlich ein Rundungsfehler auf. Dieser akkumuliert mit jeder Iteration durch die Faltung und die Reduktion. In den von mir durchgeführten Tests waren die Effekte jedoch auch mehreren tausend Faltungen zu gering, um das Ergebnis zu verfälschen. Dieser Fehler ist natürlich größer als bei kontinuierlichen Verteilungen, da die Anzahl der zu Beschreibung notwendigen Parametern deutlich größer ist.

5.3.2 Mathematische Eleganz

Ein etwas subjektiver Nachteil ist die geringere mathematische Eleganz. Schnittpunkte, Erwartungswerte und Standardabweichungen müssen mit Hilfe aller Einzelwerte berechnet werden; die Anzahl der Parameter ist abhängig von der gewünschten Genauigkeit.

Literaturverzeichnis

[Berger et al., 2011] Berger, A., 0001, A. G., Müller-Hannemann, M., and Ostrowski, M. (2011). Stochastic delay prediction in large train networks. In Caprara, A. and Kontogiannis, S. C., editors, *ATMOS*, volume 20 of *OASICS*, pages 100–111. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[Chawla and Roughgarden, 2006] Chawla, S. and Roughgarden, T. (2006). Single-source stochastic routing. 4110:82–94.

[Nolan, 2007] Nolan, J. P. (2007). *Stable Distributions - Models for Heavy Tailed Data*. Birkhäuser, Boston. In progress, Chapter 1 online at academic2.american.edu/~jpnolan.