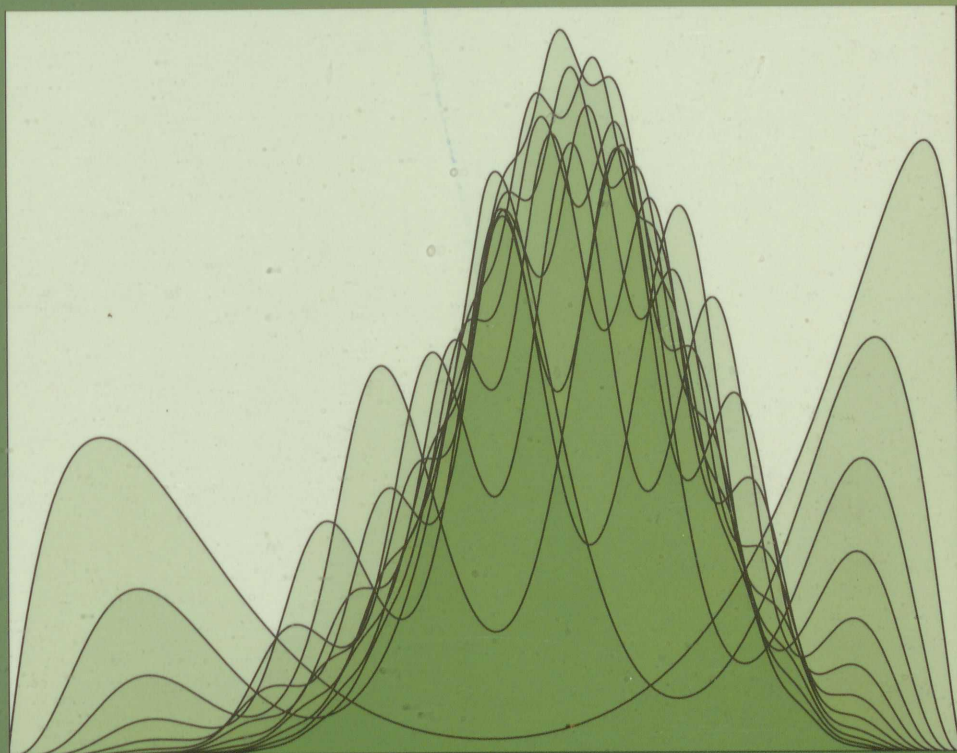


Marcin Korzeń

Obliczenia numeryczne na zmiennych losowych



Szczecin 2013

141 119-124

W6apr.

Zachodniopomorski Uniwersytet Techniczny w Szczecinie

Spis treści

Data symboli

Procedura

1. Wprowadzenie

1.1. Cel i zakres

1.2. Definicje

1.3. Zakres i granice

1.4. Terminologia

1.5. Skróty i symbole

Obliczenia numeryczne na zmiennych losowych

2.1. Wprowadzenie

2.2. Definicje

2.3. Zakres i granice

2.4. Terminologia

2.5. Skróty i symbole

2.6. Wprowadzenie

2.7. Definicje

2.8. Zakres i granice

2.9. Terminologia

2.10. Skróty i symbole

2.11. Wprowadzenie

2.12. Definicje

2.13. Zakres i granice

2.14. Terminologia

2.15. Skróty i symbole

2.16. Wprowadzenie

2.17. Definicje

2.18. Zakres i granice

2.19. Terminologia

2.20. Skróty i symbole

Szczecin 2013

Recenzenci

PROF. DR HAB. INŻ. OLGIERD HRYNIEWICZ

DR HAB. RYSZARD KOZERA

Opracowanie redakcyjne

KATARZYNA MITAN



W.141119

WYDANO ZA ZGODĄ

REKTORA ZACHODNIOPOMORSKIEGO UNIWERSYTETU TECHNOLOGICZNEGO
W SZCZECINIE

ISBN 978-83-7663-126-4

Wydawnictwo Uczelniane Zachodniopomorskiego Uniwersytetu Technologicznego w Szczecinie
70-311 Szczecin, al. Piastów 50, tel. 91 449 47 60, e-mail: wydawnictwo@zut.edu.pl
Druk PPH „ZAPOL” Dmochowski, Sobczyk Sp.j., 71-062 Szczecin, al. Piastów 42, tel. 91 434 10 21
e-mail: zarzad@zapol.com.pl

2.194603

Spis treści

Lista symboli	7
Przedmowa	9
1. Wprowadzenie	11
1.1. Obliczenia probabilistyczne oraz wnioskowanie	11
1.2. Cel badań	15
1.3. Arytmetyka zmiennych losowych i propagacja niepewności	16
1.4. Przegląd stosowanych rozwiązań	17
1.4.1. Arytmetyka interwałowa	17
1.4.2. Propagacja błędów	17
1.4.3. Histogramy i dyskretyzacja	18
1.4.4. Arytmetyka rozmyta	18
1.4.5. Arytmetyka zmiennych losowych	19
1.5. Istniejące biblioteki i oprogramowanie do obliczeń na zmiennych niepewnych	19
1.6. Wnioskowanie	21
2. Operacje na zmiennych losowych niezależnych	23
2.1. Wstęp	23
2.2. Operacje arytmetyczne na zmiennych losowych	24
2.3. Funkcje zmiennych losowych	25
2.4. Operacje arytmetyczne z użyciem funkcji kawałkami gładkich	26
2.4.1. Funkcje kawałkami gładkie	26
2.4.2. Dodawanie i odejmowanie	27
2.4.3. Mnożenie	29
2.4.4. Dzielenie	30
2.4.5. Funkcje zmiennych losowych	31
2.5. Minimum i maksimum, rozkłady statystyk pozycyjnych	31
2.6. Zmienne dyskretne i mieszane	33
2.7. Wybór sposobu reprezentacji	35
2.7.1. Transformacje i momenty	35
2.7.2. Numeryczna reprezentacja funkcji gęstości i dystrybuanty	36
2.8. Klasy rozkładów zamknięte ze względu na operacje arytmetyczne i potęgowanie	37

3. Obliczenia na zmiennych losowych zależnych	43
3.1. Wstęp	43
3.2. Wnioskowanie probabilistyczne	44
3.2.1. Sformułowanie zagadnienia	44
3.2.2. Stosowane rozwiązania	44
3.3. Wielowymiarowe modele zależności	45
3.3.1. Wielowymiarowy rozkład normalny	45
3.3.2. Kopuły	46
3.3.3. Kopuły archimedesowe	49
3.3.4. Próba uporządkowana, łączny rozkład dwóch statystyk pozycyjnych	51
3.4. Dwuwymiarowe miary zależności pomiędzy zmiennymi	52
3.5. Obliczenia na zmiennych zależnych	54
3.5.1. Twierdzenie o zamianie zmiennych	54
3.5.2. Wyrażenia algebraiczne zawierające dwie zmienne losowe	56
3.5.3. Skrajne zależności pomiędzy zmiennymi, kopuły M i W	59
3.5.4. Ograniczenia na dystrybuantę wyniku	61
4. Metody numeryczne stosowane w obliczeniach probabilistycznych	63
4.1. Wstęp	63
4.2. Interpolacja i aproksymacja	64
4.2.1. Interpolacja barycentryczna	64
4.2.2. Wymierna interpolacja barycentryczna	66
4.3. Obliczenia z użyciem wielomianów Czebyszewa	67
4.3.1. Związki z szeregami Fouriera	67
4.3.2. Przyrostowy algorytm interpolacji	71
4.3.3. Całkowanie szeregów Czebyszewa	74
4.3.4. Różniczkowanie szeregów Czebyszewa	74
4.3.5. Miejsca zerowe, ekstrema	75
4.4. Całkowanie numeryczne	77
4.4.1. Całki na przedziale skończonym	77
4.4.2. Przedziały nieskończone, funkcje z osobliwościami	79
4.5. Dokładność aproksymacji	81
4.5.1. Rezultaty teoretyczne	81
4.5.2. Dokładność obliczeń zmiennopozycyjnych	82
4.5.3. Ograniczenia formuły barycentrycznej	84
4.6. Aproksymacja funkcji gęstości, funkcje kawałkami gładkie	85
4.6.1. Funkcje na przedziale skończonym	85
4.6.2. Funkcje na przedziałach nieskończonych, funkcje z osobliwościami	86
4.6.3. Aproksymacja w klasie funkcji $\mathcal{F}$	89
4.7. Pakiet chebfun	89
4.8. Obliczenia numeryczne w pakiecie PaCAL	91

4.9.	Przykłady numeryczne	94
4.9.1.	Funkcje ciągłe na przedziale domkniętym	94
4.9.2.	Funkcje z osobliwościami, funkcje na przedziałach ograniczonych	98
4.9.3.	Funkcje z ciężkimi ogonami	102
5.	Projekt PaCAL	105
5.1.	Cel i główne założenia projektu	105
5.2.	Instalacja i pierwsze uruchomienie	106
5.3.	Obliczenia na zmiennych niezależnych	107
5.3.1.	Szczegóły implementacyjne	107
5.3.2.	Rozkłady jednowymiarowe zdefiniowane w PaCAL	108
5.3.3.	Metody klasy Distr	113
5.3.4.	Arytmetyka zmiennych losowych niezależnych	115
5.3.5.	Zmienne dyskretne i mieszane	116
5.3.6.	Rozkłady warunkowe	117
5.3.7.	Estymacja parametrów	118
5.4.	Semantyka	118
5.5.	Praca z biblioteką PaCAL	119
5.6.	Rozkłady wielowymiarowe, obliczenia na zmiennych zależnych	120
6.	Przykłady i zastosowania	123
6.1.	Testowanie dokładności	123
6.2.	Przykłady numeryczne	126
6.2.1.	Środowisko testowe	126
6.2.2.	Porównanie z pakietami obliczeń symbolicznych	126
6.2.3.	Porównanie z pakietem chebfun	128
6.2.4.	Rozwiązania teoretyczne	130
6.2.5.	Rozkłady niecentralne	132
6.3.	Możliwości i zastosowania pakietu PaCAL	135
6.3.1.	Wnioskowanie statystyczne	135
6.3.2.	Pomiary wielkości fizycznych	136
6.3.3.	Agregacja ocen eksperckich	138
6.4.	Obliczenia na zmiennych zależnych	143
6.4.1.	Rozstęp z próby, rozstęp międzykwantylowy	143
6.4.2.	Skrajne możliwe wartości sumy zmiennych losowych	144
7.	Podsumowanie	147
	Bibliografia	149
	Indeks	157
	Summary	161
	Zusammenfassung	163

Lista symboli

$\mathcal{A}[a, b]$	–	zbiór funkcji analitycznych na przedziale $[a, b]$
$Bernoulli(p)$	–	rozkład Bernoulliego (dwupunktowy)
$Beta(\alpha, \beta)$	–	rozkład beta
$Binom(n, p)$	–	rozkład dwumianowy
$\mathbb{C}$	–	liczby zespolone
$C[a, b]$	–	zbiór funkcji ciągłych na $[a, b]$
$C^k[a, b]$	–	zbiór funkcji k -krotnie różniczkowalnych w sposób ciągły na $[a, b]$
$C(\gamma, c)$	–	rozkład Cauchy'ego
CDF_X	–	dystrybuanta zmiennej losowej X
$CDF_X(x)$	–	wartość dystrybuanty zmiennej losowej X w punkcie x
$Chi2(df)$	–	rozkład χ^2
$E(\lambda)$	–	rozkład wykładniczy
$\mathcal{F}$	–	klasa rozkładów zamknięta ze względu na operacje probabilistyczne
$\mathcal{F}(f)$	–	przekształcenie Fouriera funkcji f
$f \sim g$	–	$f(x) \sim g(x)$, $x \rightarrow x_0$ oznacza $f(x)/g(x) \rightarrow 1$ dla $x \rightarrow x_0$
$f, g, \dots$	–	funkcje gęstości zmiennych losowych
f'	–	pochodna funkcji f
$f^{(i)}$	–	i -ta pochodna funkcji f , $f^{(0)} = f$, $f^{(i)} = (f^{(i-1)})'$, $i = 1, 2, \dots$
$F, G, \dots$	–	dystrybuanty zmiennych losowych
$F(df_1, df_2)$	–	rozkład F Snedecora
$Fr(\alpha, s, m)$	–	rozkład Frécheta
$Gamma(k, \theta)$	–	rozkład gamma
$Gu(\mu, \sigma)$	–	rozkład Gumbela
$L(\lambda)$	–	rozkład Laplace'a
$L(c, x_{\min})$	–	rozkład Levy'ego
$\mathbb{N}$	–	liczby naturalne: $1, 2, \dots$
$N(\mu, \sigma)$	–	rozkład normalny
$NCB(\alpha, \beta, \lambda)$	–	niecentralny rozkład beta
$NCF(df_1, df_2, \lambda)$	–	niecentralny rozkład F
$NCT(df, \mu)$	–	niecentralny rozkład T

$NCX(df, \lambda)$	–	niecentralny rozkład chi-kwadrat
$\mathcal{O}$	–	$f(n) = \mathcal{O}(g(n))$, jeżeli $f(n) \leq Cg(n)$
$P(\alpha, x_{\min})$	–	rozkład Pareto
PDF_X	–	funkcja gęstości zmiennej X
$PDF_X(x)$	–	wartość funkcji gęstości zmiennej losowej X w punkcie x
$\mathbb{R}$	–	liczby rzeczywiste
$T(df)$	–	rozkład t -Studenta
$Trapz(a, b, c, d)$	–	rozkład trapezowy
$U(a, b)$	–	rozkład jednostajny
$We(k, \lambda)$	–	rozkład Weibulla
$X, Y, \dots$	–	zmienne losowe
$X = Y$	–	zmienne losowe X i Y są identyczne co do realizacji
$X \sim f, F$	–	zmienna losowa X o funkcji gęstości f oraz dystrybuancie F
$X \sim Y$	–	zmienna losowa X ma taki sam rozkład jak zmienna Y
$X_{(i)}$	–	i -ta statystyka pozycyjna
$\mathbb{Z}$	–	liczby całkowite
$\chi_{[a,b]}$	–	funkcja charakterystyczna przedziału $[a, b]$
Σ'	–	$\sum_{k=0}^m a_k = \frac{1}{2}a_0 + a_1 + \dots + a_n$
Σ''	–	$\sum_{j=0}^m f_j = \frac{1}{2}f_0 + f_1 + \dots + \frac{1}{2}f_n$
$\ f\ _1$	–	norma L^1 funkcji f
$\ f\ _\infty$	–	norma L^∞ funkcji f
$k n$	–	relacja podzielności, k dzieli n

Przedmowa

Monografia dotyczy obliczeń arytmetycznych na zmiennych losowych. Znaczne jej fragmenty powstały w efekcie prac nad projektem finansowanym przez Ministerstwo Nauki i Szkolnictwa Wyższego nr N N516 068537, pod tytułem „**PaCAL** – biblioteka obliczeń arytmetycznych na zmiennych losowych”. Kierownikiem projektu był dr hab. inż. Szymon Jaroszewicz, a głównym wykonawcą autor monografii. W szczególności dotyczy to obliczeń arytmetycznych na zmiennych losowych niezależnych oraz wczesnej implementacji wnioskowania w sieciach bayesowskich. Wynikiem realizacji tego projektu jest biblioteka dostępna pod adresem <http://pacal.sf.net> oraz prace [51, 68].

Zakres monografii dotyczy części projektu, za którą w znacznym stopniu odpowiadał autor monografii, czyli przede wszystkim obliczeń arytmetycznych na zmiennych losowych niezależnych oraz obliczeń w prostych modelach zmiennych zależnych. Część dotycząca zmiennych niezależnych została szczegółowo opisana w rozdz. 2, a w rozdz. 3 omówiono takie modele zmiennych zależnych, jak wielowymiarowy rozkład normalny czy modele zależności z użyciem kopuł. Opisano także obliczenia na zmiennych skrajnie skorelowanych oraz wyznaczanie dystrybuant ograniczających możliwą wartość wyniku przy założeniu jedynie rozkładów brzegowych zmiennych. Ogólne wnioskowanie probabilistyczne w modelach graficznych, za które to wnioskowanie odpowiadał w większym stopniu dr hab. inż. Szymon Jaroszewicz, wykracza poza zakres monografii.

W rozdziale 4 omówiono szczegółowo dostępną implementację projektu **PaCAL** od strony metod numerycznych. Przedstawiono tam przede wszystkim odpowiednie metody interpolacji i całkowania, w tym interpolację barycentryczną wykorzystującą węzły Czebyszewa oraz metody numeryczne oparte na szeregach Czebyszewa. Zwrócono także uwagę na obliczenia z użyciem funkcji mających osobliwości oraz funkcji określonych na przedziałach nieskończonych.

W rozdziale 5 opisano implementację praktyczną projektu **PaCAL**, a w rozdz. 6 – szereg zastosowań przedstawiających uniwersalność oraz wysoką dokładność przyjętych rozwiązań numerycznych. Prezentowany zakres zastosowań obejmuje procedury wniosko-

wania statystycznego i testowanie hipotez, pomiary czy agregacje ocen eksperckich. Przedstawiono też bardziej złożone przykłady obejmujące przypadek zmiennych zależnych.

Własności teoretyczne operacji arytmetycznych na niezależnych zmiennych losowych oraz wstępną wersję pakietu **PaCAL** (v1.0 – obejmującą arytmetykę zmiennych niezależnych ciągłych) opisano w pracy [51]. W szczególności przedstawiono tam twierdzenia o zamkniętości rodziny rozkładów ze względu na operacje arytmetyczne oraz potęgi zmiennych losowych o wykładnikach rzeczywistych. Stan biblioteki w wersji 1.1 (obejmujący dodatkowo zmienne dyskretne i mieszane oraz obliczenia w prostych modelach zmiennych zależnych) z punktu widzenia użytkownika wraz z pewną liczbą przykładów przedstawiono po raz pierwszy w pracy [68]. W stosunku do tych pozycji w niniejszej monografii dodano: przegląd technik obliczeniowych na zmiennych niezależnych (rozdz. 2), szczegółowy opis stosowanych metod i technik numerycznych dotyczących głównie całkowania i aproksymacji funkcji opartych na wielomianach Czebyszewa oraz transformacjach zmiennych (rozdz. 4), dokładny opis obliczeń w prostych modelach zmiennych zależnych, w tym również przypadek zmiennych o znanym modelu zależności oraz zmiennych o nieznannej korelacji (rozdz. 3). Ponadto w rozdz. 6 przedstawiono szereg nowych eksperymentów oraz zastosowań, jak na przykład agregacja i ocena oszacowań ekspertów. Większość eksperymentów i przykładów (o ile nie zaznaczono inaczej w tekście) wykonano na nowo.

Marcin Korzeń

mkorzen@wi.zut.edu.pl

1. Wprowadzenie

1.1. Obliczenia probabilistyczne oraz wnioskowanie

Monografia dotyczy automatyzacji obliczeń numerycznych na zmiennych losowych, a szczególnie opracowania procedur numerycznych wspomagających obliczenie zadań typowych dla rachunku prawdopodobieństwa. Stosując podejście probabilistyczne w praktyce, przeprowadzamy pewne rodzaje wnioskowań probabilistycznych czy statystycznych. W takim przypadku niejawnie zakładamy, że mamy do czynienia ze zmiennymi o charakterze losowym, to znaczy takim, którego wyniku nie da się z góry przewidzieć. Niemniej jesteśmy w stanie w pewnym stopniu przewidzieć charakter zaburzeń losowych, opierając się na pewnej liczbie podobnych czy wcześniejszych obserwacji.

Wykonywanie obliczeń na zmiennych określonych w sposób nieprecyzyjny ma duże znaczenie oraz pojawia się w wielu zastosowaniach zarówno praktycznych, jak i teoretycznych. W praktyce istnieje wiele formalizmów dotyczących opisu niepewności. Wydaje się, że najlepiej zbadany od strony teoretycznej jest rachunek prawdopodobieństwa. Jednak istnieje również szereg nieklasycznych podejść do modelowania niepewności, jak logika i arytmetyka rozmyta [127], teoria Dempstera-Shafera [109] czy teoria luk informacyjnych [7]. W niniejszej pracy będzie położony szczególny nacisk na metody probabilistyczne, natomiast odniesienie do odmiennych koncepcji zostało ograniczone do zakresu, w jakim są one zbieżne (lub podobne) z metodami probabilistycznymi.

Metodologia obliczeń probabilistycznych czy obliczeń na zmiennych losowych jest dobrze znana i w pewnym zakresie zostanie przedstawiona w kolejnych rozdziałach. Zwykle wnioskowanie probabilistyczne polega na wyznaczeniu rozkładu pewnej interesującej nas zmiennej lub zmiennych na podstawie danego modelu probabilistycznego – czyli takiego, w którym wszystkie potrzebne zależności i rozkłady brzegowe są dane. Z technicznego punktu widzenia szukany rozkład znajdujemy przez pewną liczbę całkowań, w ogólności funkcji wielowymiarowych. Pewne szczególne przypadki wnioskowań probabilistycznych, jak arytmetyka zmiennych losowych niezależnych, można wykonywać w miarę efektywnie – wymagane jest liczenie jedynie całek jednokrotnych. Jednak nawet w takim przypadku utrzymanie wysokiej dokładności obliczeń jest pewnym wyzwaniem. Należy być świadomym, że funkcja gęstości może mieć osobliwości, nieciągłości czy być określona na przedziale nieskończonym. W ogólności również nie da się uniknąć liczenia całek wielo-

wymiarowych nawet dla prosto wyglądających modeli, co powoduje, że problem ten staje się złożony numerycznie.

Zasadniczym zadaniem obliczeń probabilistycznych jest wyznaczenie rozkładu pewnej zmiennej lub zmiennych w ustalonym modelu probabilistycznym. Główne operacje wykorzystywane w trakcie obliczeń probabilistycznych to: zamiana zmiennych, marginalizacja (lub eliminacja zmiennych) oraz warunkowanie. Dwie ostatnie operacje wymagają w ogólności całkowania, natomiast przy zamianie zmiennych wymagane jest różniczkowanie oraz rozwiązywanie równań. W przypadku, gdy zasadniczą częścią obliczeń jest warunkowanie na pewnej próbie danych, proces obliczeniowy nosi często nazwę wnioskowania [96, 97]. Zwykle wnioskowanie wymaga wykonania sekwencji wyżej wymienionych prostych operacji. Z tego powodu proces ten trudno jest zautomatyzować. W praktyce dla rozwiązywania zagadnienia wnioskowania probabilistycznego jest stosowanych kilka różnych rozwiązań, które zostaną omówione poniżej.

Rozwiązania teoretyczne

Historycznie pierwsze są rozwiązania teoretyczne. Rozwiązania takie podane są dla znacznej liczby przypadków mniej lub bardziej szczególnych. W praktyce znaczna liczba procedur i testów statystycznych korzysta z takich rozwiązań. Na przykład przez bezpośrednie całkowanie można sprawdzić, że suma niezależnych zmiennych losowych o standaryzowanym rozkładzie normalnym ma rozkład normalny o wartości średniej zero oraz wariancji n lub suma kwadratów niezależnych zmiennych losowych o standaryzowanym rozkładzie normalnym ma rozkład χ^2 o liczbie stopni swobody zależnej od liczby składników. Jednak nawet w prostych przypadkach odpowiednie całki nie wyrażają się za pomocą funkcji elementarnych, co powoduje konieczność korzystania z coraz bardziej skomplikowanych funkcji specjalnych. Widać to na przykładzie niecentralnego rozkładu χ^2 , liczonego jako suma kwadratów przesuniętych, niezależnych rozkładów normalnych o jednakowej wariancji [113, podrozdz. 9.2.7]. Gęstość niecentralnego rozkładu χ^2 wyraża się za pomocą zmodyfikowanej funkcji Bessela lub funkcji hipergeometrycznej [1, s. 942]. Funkcje te dane są za pomocą szeregów i wymagają osobnych implementacji lub stabilizowania. W rozdziale 6 przedstawiono przykłady numeryczne pokazujące, że implementacje rozkładów niecentralnych wykorzystujące funkcje hipergeometryczne są trudne w implementacji, co widać w istniejących bibliotekach statystycznych.

Obecnie najbardziej zaawansowanym (z obliczeniowego punktu widzenia) rozwinięciem podejścia teoretycznego są rozwiązania oparte na systemach algebry komputerowej CAS (ang. *Computer Algebra System*), jak na przykład pakiet **APPL** [39] zaimplementowany w programie Maple czy funkcje probabilistyczne programu Mathematica. Systemy algebry komputerowej dobrze radzą sobie z przykładami, które można scałkować symbolicznie. Jednak ogólne rozwiązania dla tego typu systemów są trudne do uzyskania. Ponadto szybkość tego typu systemów, zwłaszcza przy wykonywaniu większej liczby operacji,

również pozostawia wiele do życzenia. Do rozwiązań tych będzie odniesienie w dalszych częściach pracy (podrozdz. 6.2.5).

Modele, w których możliwe jest dokładne wnioskowanie probabilistyczne

Istnieje wiele modeli probabilistycznych, w których wnioskowanie¹ można wykonywać w sposób dokładny. Przykładem takich modeli są sieci bayesowskie skończone ze zmiennymi dyskretnymi. W modelach tych mamy możliwość wnioskowania w sposób dokładny, to jest znajdowania rozkładów dowolnych zmiennych, pod warunkiem że ustalone są wartości niektórych z pozostałych zmiennych. Modele te często są wykorzystywane w praktyce. J. Pearl w swoich książkach [96, 97] praktycznie zawsze używa symbolu sumy, podkreślając fakt korzystania ze zmiennych dyskretnych (zamiast ogólnie całek), podobnie mamy w podręcznikach F. Jensena [53, 54]. Pomijamy w tym miejscu przypadek samej złożoności obliczeniowej procedury wnioskowania w sieciach bayesowskich, jednak wiadomo, że wnioskowanie w sieciach bayesowskich jest w ogólnym przypadku NP-trudne [21].

Kolejny przypadek, w którym pewne rodzaje wnioskowań można wykonywać dokładnie (drugi obok przypadku skończonego dyskretnego), dotyczy zmiennych ciągłych opisanych łącznym wielowymiarowym rozkładem normalnym [11, rozdz. 8], co wynika z własności rozkładu normalnego. To znaczy rozkłady brzegowe i warunkowe wyznaczone z wielowymiarowego rozkładu normalnego są rozkładami normalnymi, również suma niezależnych zmiennych o rozkładzie normalnym daje w wyniku przeskalowany rozkład normalny.

Podejście numeryczne

Podejście numeryczne do obliczeń na zmiennych losowych jest stosowane od dość dawna. Zasadniczą trudnością w takim podejściu jest całkowanie oraz przechowywanie obliczeń pośrednich i wyniku końcowego. Całkowania numerycznego nie da się uniknąć nawet w najprostszych przypadkach. Widać to na przykładzie dystrybucyj rozkładu normalnego i związanej z nią funkcji błędu (ang. *erf*, *error function*). Funkcja ta nie wyraża się w postaci skończonej za pomocą funkcji elementarnych i standardowym rozwiązaniem jest tu numeryczne stabilizowanie lub interpolacja rozkładu wynikowego. Rozwiązania numeryczne zwykle dotyczą szczególnych przypadków, związanych zazwyczaj z konkretną procedurą wnioskowania statystycznego. Często praktyką w tym zakresie jest wykorzystywanie rozwinięć teoretycznych danych szeregami oraz numeryczna aproksymacja takich szeregów.

Bardziej zaawansowane podejścia numeryczne polegają na zastosowaniu zamkniętych reprezentacji oraz wykorzystaniu własności takich reprezentacji. Na przykład jeżeli ograniczymy się do arytmetyki zmiennych niezależnych oraz wykorzystamy momenty (lub

¹ Spotykanym czasami synonimem jest inferencja (ang. *inference*).

kumulanty) do reprezentacji zmiennych losowych, jesteśmy w stanie łatwo wykonywać operację dodawania (odejmowania) [66, rozdz. 2], dodając odpowiednie ciągi momentów (kumulant). Odtwarzanie rozkładu na podstawie kumulant prowadzi do szeregów Edgewortha [66, rozdz. 2]. Podejście takie, będące metodą ogólną, jest jednak ograniczone do funkcji gęstości mających momenty (kumulanty) wszystkich rzędów. Ponadto dzielenie dla reprezentacji opartych na momentach nie może być wykonane w sytuacji ogólnej. Dobrze znanym przykładem są tu dwie zmienne niezależne o wspólnym standaryzowanym rozkładzie normalnym. Momenty rozkładu normalnego są dobrze określone, natomiast iloraz takich zmiennych daje w wyniku rozkład Cauchy'ego [28], który nie ma średniej ani momentów wyższego rzędu.

Inne stosowane podejścia wykorzystują wielomiany Laguerre'a czy Hermite'a oraz własności transformaty Fouriera do liczenia splotów (co odpowiada sumie zmiennych losowych niezależnych) czy transformacji i splotu Melina (co odpowiada mnożeniu zmiennych niezależnych) [15, 19, 124]. Problemem ponownie jest tu wykonywanie dzielenia. Bardziej złożony aparat H -funkcji jest wyłożony w pracy [113]. H -funkcje są uogólnieniem funkcji hipergeometrycznych, które dane są za pomocą odpowiednich szeregów. Dzięki H -funkcjom możemy dostać klasę rozkładów zamkniętą ze względu na mnożenie i dzielenie. Wadą jest w tym przypadku brak zamkniętości tej klasy funkcji gęstości ze względu na dodawanie i odejmowanie [113]. Praktyka pokazuje również, że posługiwanie się H -funkcjami, podobnie jak szeregami hipergeometrycznymi, jest trudne numerycznie.

Metody Monte Carlo

Dalsze metody rozwiązywania zagadnienia wnioskowania probabilistycznego są w swojej idei przybliżone. Podstawową metodą ogólną, najszerzej stosowaną w praktyce, jest grupa metod znanych pod wspólną nazwą Monte Carlo [81]. W metodach Monte Carlo zamiast liczenia całek bezpośrednio w sposób numeryczny używa się odpowiednich schematów próbkowania, jak próbkowanie Gibbsa czy ważnościowe [2, 78]. Podejścia te obecnie mają ugruntowaną pozycję. Istnieje wiele praktycznych implementacji umożliwiających wnioskowanie w złożonych modelach sieci bayesowskich czy innych modelach probabilistycznych. Przykładami systemów wnioskowania wykorzystujących tego typu podejście są: **BUGS** [76, 112], **JAGS** [102], **PyMC** [95], **SMILE/Ginie** [24]. Głównymi zaletami metod opartych na próbkowaniu są względnie prosta implementacja (choć niektóre typy wnioskowania, jak warunkowanie, mogą być bardziej złożone) oraz szybkość działania. Główną wadą jest natomiast dokładność obliczeń, która skaluje się z pierwiastkiem z rozmiaru użytej próby [126].

W tym miejscu należy zwrócić uwagę, że istnieje pewna różnica pomiędzy dokładnością obliczeń numerycznych a dokładnością metod typu Monte Carlo. Metody numeryczne, mimo że również są przybliżone, to jednak umożliwiają zwykle precyzyjną kontrolę dokładności obliczeń (jak na przykład odrzucenie reszty szeregu), a prawidłowa

implementacja daje zwykle błąd obliczeń porównywalny z precyzją zapisu liczb zmiennopozycyjnych. Natomiast gwarancje dokładności metod typu Monte Carlo opierają się głównie na nierównościach typu Hoeffdinga czy Chernoffa [3]. Tego typu oszacowania pozwalają dobrać rozmiar próby, aby jedynie *prawdopodobieństwo* odchylenia od rezultatu teoretycznego o pewną ustaloną wartość było na zadanym poziomie. Dla porównania rozważmy całkę z funkcji sinus na przedziale $[0, 1]$. Stosując metodę Clenshawa–Curtisa, wystarczy obliczyć (w tym przypadku) wartości funkcji jedynie w 15 specjalnie wybranych węzłach, aby uzyskać dokładność równą precyzji maszynowej. Dokładność tego rzędu nie jest możliwa do uzyskania z użyciem metody Monte Carlo (przy rozsądnym rozmiarze próby). Tego typu przykład pokazuje przewagę odpowiednio zaprojektowanej metody numerycznej nad metodami typu Monte Carlo. Jednak dla funkcji wielu zmiennych sytuacja wygląda trochę inaczej. Liczenie całek wielowymiarowych jest problemem obliczeniowym, analogiczne schematy numeryczne wymagają zwykle liczby węzłów wykładniczo zależnej od liczby wymiarów. W takich przypadkach podejścia typu Monte Carlo, stosujące odpowiednie schematy próbkowania zależne od całkowanej funkcji, stają się znacznie bardziej konkurencyjne.

1.2. Cel badań

Obecnie nie istnieją ogólne narzędzia pozwalające wykonywać w dużej ogólności obliczenia na zmiennych losowych niezależnych w sposób numerycznie dokładny, oparte na jednolitych reprezentacjach numerycznych. Podobna uwaga dotyczy również metod ogólnego wnioskowania probabilistycznego. W zasadzie jedyną ogólną obecnie stosowaną techniką w obliczeniach probabilistycznych są metody typu Monte Carlo [2]. Istniejące rozwiązania numeryczne odnoszą się zwykle do przypadków szczególnych. Pewne rozwiązania ogólne dotyczące obliczeń na zmiennych losowych niezależnych istnieją w obrębie obliczeń symbolicznych [39, 125], jednak zakres ich stosowalności jest ograniczony możliwościami obliczeń symbolicznych (zwłaszcza całkowania). Taki stan rzeczy był jedną z motywacji do rozpoczęcia prac nad projektem **PaCAL** [52]. Podstawowa teza badawcza, na której były oparte badania, mówi, że *możliwe jest opracowanie efektywnego mechanizmu obliczeń probabilistycznych na zmiennych losowych niezależnych oraz wykonywanie innych obliczeń probabilistycznych z użyciem numerycznie dokładnych reprezentacji zmiennych losowych*. W takim zakresie koncepcja ta jest całkowicie odmienna od powszechnie stosowanych metod.

Opisany w monografii mechanizm obliczeniowy na zmiennych losowych został praktycznie zaimplementowany w ramach projektu „**PaCAL** – biblioteka obliczeń arytmetycznych na zmiennych losowych”. Projekt ten uzasadnia praktycznie możliwość postawienia powyższej tezy. Ponadto przykłady pokazane w rozdz. 6 pokazują wysoką efektywność

proponowanych rozwiązań w stosunku do istniejących metod. Sam projekt zostanie szczegółowo opisany w rozdz. 5.

1.3. Arytmetyka zmiennych losowych i propagacja niepewności

Szczególnym przypadkiem wnioskowania probabilistycznego jest wykonywanie obliczeń deterministycznych na zmiennych losowych, jak na przykład obliczanie operacji arytmetycznych czy wyznaczanie funkcji zmiennych losowych. Zakłada się, że dana jest pewna liczba zmiennych losowych, na których wykonujemy ustalone operacje. Natomiast interesuje nas znalezienie rozkładu zmiennej losowej będącej wynikiem tych operacji. Zakładamy w tym miejscu, że interesuje nas wnioskowanie w przód, to znaczy niezawierające warunkowania. Z probabilistycznego punktu widzenia tak postawione zadanie nie jest jeszcze możliwe do rozwiązania bez podania modelu zależności pomiędzy zmiennymi. W najprostszym przypadku zakłada się, że mamy do czynienia ze zmiennymi niezależnymi, wtedy tak postawione zadanie jest dobrze określone. Obliczeniowe aspekty arytmetyki zmiennych losowych będą omówione dokładnie w rozdz. 2. W tym miejscu zostaną przedstawione spotykane metody rozwiązywania.

Wspomniany tu problem znany jest również pod nazwą propagacji niepewności. Warto zauważyć, że zadanie propagacji niepewności ma wiele praktycznych zastosowań, na przykład w teorii pomiarów przy wykonywaniu pomiarów pośrednich czy w statystyce przy wyznaczaniu rozkładów statystyk. Na przykład interesuje nas pośredni pomiar rezystancji przy bezpośrednim pomiarze napięcia i natężenia prądu, a jako wynik bierzemy iloraz. W takim przypadku dysponujemy zwykle rozkładem obu mierzonych wartości pośrednich, a interesuje nas rozkład ilorazu tych zmiennych. Rozkład ten można wyznaczyć, używając do tego celu arytmetyki zmiennych losowych. Rzecz jasna, w przypadku pomiarów częściej zamiast dokładnego rozkładu zmiennych mierzonych mamy jedynie informacje o klasie przyrządu, wtedy zmuszeni jesteśmy stosować metody przybliżone lub dodatkowe założenia.

Dla rozwiązania zagadnienia obliczeń na zmiennych losowych istnieje wiele rozwiązań mniej lub bardziej przybliżonych. W przypadku pomiarów istnieje szczegółowa teoria zwana rachunkiem błędów lub propagacją błędów (ang. *propagation of uncertainty*) [59]. W zależności od założeń na temat modelu i rodzaju niepewności jest znanych szereg różnych metod służących do propagacji niepewności, które zostaną poniżej krótko omówione.

1.4. Przegląd stosowanych rozwiązań

1.4.1. Arytmetyka interwałowa

Jedną z prostszych metod obliczeniowych na zmiennych niepewnych jest arytmetyka interwałowa [83–85]. W arytmetyce tej zmienne opisujące niepewności reprezentowane są jako przedziały liczbowe. Umożliwia ona wyznaczenie przedziału, w którym na pewno znajduje się poprawna wartość wyniku. Przedział ten poszerza się w miarę wykonywania obliczeń. Obliczenia interwałowe służą przede wszystkim kontroli dokładności obliczeń numerycznych [114, rozdz. 1], ale mogą być uznane za przykład arytmetyki zmiennych losowych², gdzie wszystkie rozkłady są jednostajne. Jest to oczywiście bardzo duże przybliżenie i w praktyce uzyskane wyniki będą znacząco odbiegać od prawdziwych rozkładów. W arytmetyce tej występuje bardziej pesymistyczne, niż to obserwujemy zwykle w praktyce, nakładanie się niepewności. Z punktu widzenia arytmetyki probabilistycznej arytmetyka interwałowa dostarcza operacji na nośnikach³ zmiennych losowych.

1.4.2. Propagacja błędów

Zwykle do wykonywania operacji arytmetycznych na wielkościach obarczonych niepewnością stosuje się metody przybliżone. Najbardziej popularną z nich jest tak zwana metoda delta i metody propagacji błędów. Metody te są oparte na pierwszych wyrazach rozwinięcia funkcji zmiennych losowych w szereg Taylora i pozwalają na przybliżone wyrażenie odchylenia standardowego wyniku. Przegląd metod tego typu można znaleźć w pracach [20, 49, 58, 59]. Czasami przybliżenia uzyskane tymi metodami mogą być bardzo dalekie od wyników rzeczywistych, gdyż rzeczywiste rozkłady mogą bardzo znacząco odbiegać od rozkładu normalnego, a nawet być rozkładami wielomodalnymi. Generalnie metody te lepiej aproksymują wynik, jeżeli iloraz średniej i wariancji jest duży (mała zmienność). Metody te opierają się zasadniczo na przybliżeniu funkcji reprezentującej działanie szeregiem Taylora. Wartość oczekiwaną wyniku możemy wyrazić na podstawie momentów:

$$E(f(X)) \approx f(\mu_X) + \frac{f''(\mu_X)}{2} \sigma_X^2.$$

² W tym znaczeniu w literaturze używa się również terminu „arytmetyka probabilistyczna” lub „algebra zmiennych losowych” (por. tytuły prac [113, 124]), przy czym termin „arytmetyka” wydaje się tu bardziej odpowiedni, gdyż mamy tu na myśli jedynie wykonywanie działań na zmiennych losowych. Termin „algebra” sugerowałby spełnienie pewnych własności algebraicznych, jak istnienie zera czy jedności oraz możliwość rozwiązywania równań. Mimo że działania na zmiennych losowych spełniają pewne własności algebraiczne (na przykład suma zmiennych niezależnych jest łączna i przemienne), to jednak kwestia elementów neutralnych czy przeciwnych jest bardziej problematyczna.

³ Nośnik (ang. *support*) jest to przedział, na którym jest określona zmienna lub zbiór argumentów, dla których wartość funkcji gęstości jest dodatnia.



1.4.3. Histogramy i dyskretyzacja

W przypadku obliczeń na zmiennych losowych ciągłych w zasadzie spotyka się dwa ogólne podejścia:

- *próbkowanie* – na tej koncepcji opierają się metody typu Monte Carlo;
- *dyskretyzację* – na zmiennych dyskretnych obliczenia można wykonywać dokładnie, tego typu podejście można spotkać w większości spotykanych implementacji.

Połączony Komitet ds. Wytycznych w Metrologii (ang. *JCGM Joint Committee for Guides in Metrology*), działający pod kierownictwem dyrektora Międzynarodowego Biura Miar, w swoim przewodniku „Guide to the expression of uncertainty in measurement” [59] podaje sposoby postępowania z niepewnościami w metrologii. Opisano tam szeroko zakres stosowanych i zalecanych metod w rachunku błędów. Jedną z głównych zalecanych metod ogólnego typu (obok metody delta) jest metoda Monte Carlo [60], polegająca na próbkowaniu rozkładów wejściowych i przekształceniu ich przez zadaną operację, tak aby otrzymać przybliżony rozkład wyniku. Na podstawie uzyskanych histogramów można oszacować przeciętną wartość pomiaru, przedział ufności itp. Postępowanie takie jest szczególnym przypadkiem metod wnioskowania probabilistycznego opartych na próbkowaniu zastosowanym do propagacji niepewności pomiarowych. W pracy [75] przedstawiono koncepcję arytmetyki probabilistycznej opartą na histogramach.

1.4.4. Arytmetyka rozmyta

Teoria zbiorów rozmytych opracowana przez L. Zadeha [127] ma za zadanie dostarczać mechanizmów obliczeniowych w przypadku informacji niepewnych oraz nieprecyzyjnie określonych. Jedną z poddziedzin teorii systemów rozmytych jest arytmetyka rozmyta określona jako aparat obliczeniowy na liczbach rozmytych, które można traktować jako pewne uogólnienie niepewności interwałowych. Ogólna technika obliczeń na liczbach rozmytych opiera się na zasadzie rozszerzenia Zadeha [77, 98]. Ogólna idea obliczeń na zmiennych rozmytych jest podobna do obliczeń na zmiennych losowych niezależnych. Jednak w miejsce sumowania (czy całkowania) stosujemy odpowiednie S -normy (jak supremum). Szczegółowy opis arytmetyki rozmytej można znaleźć w pracach [25, 62, 63, 77, 98]. Określenie wzajemnych związków teorii zbiorów rozmytych i rachunku prawdopodobieństwa wykracza zdecydowanie poza zakres monografii. Wspomnieć można tylko, że sam Zadeh uważa obie teorie za komplementarne.

Z praktycznego punktu widzenia niepewność opisana liczbą rozmytą nie ma tak intuicyjnej interpretacji, jak niepewność opisana zmienną losową. Wartość przeciętna czy przedział ufności są dobrze zrozumiałe dla praktyka. Część dyskusji na temat wzajemnych relacji pomiędzy logiką rozmytą a teorią prawdopodobieństwa można znaleźć na stronach

dyskusyjnych inicjatywy Uniwersytetu Berkeley⁴ „The Berkeley Initiative in Soft Computing”. Natomiast bardziej probabilistyczny punkt widzenia można znaleźć w artykule [16].

Istnieje wiele alternatywnych podejść do arytmetyki rozmytej. Na przykład w pracy [69] opisano podejście oparte na skierowanych liczbach rozmytych. Wyniki obliczeń na takich liczbach różnią się od tych uzyskiwanych za pomocą zasady rozszerzenia Zadeha. W tej konstrukcji silny nacisk jest położony na zachowanie własności algebraicznych w trakcie obliczeń. Skierowane liczby rozmyte, mimo że nie mają również jasnej interpretacji fizycznej, mają szereg ciekawych własności algebraicznych, jak istnienie zera i jedności.

Należy też zauważyć, że wyniki arytmetyki rozmytej różnią się od formalizmu probabilistycznego. Zaletą metod probabilistycznych jest umożliwienie bardziej precyzyjnego opisu zależności pomiędzy zmiennymi (korelacja, rozkłady łączne, modele graficzne itp.), ponadto mają jasną interpretację fizyczną. Rodzaj zależności pomiędzy zmiennymi losowymi jest zwykle jedną z danych wejściowych lub założeń w modelu probabilistycznym.

1.4.5. Arytmetyka zmiennych losowych

Arytmetyka probabilistyczna [113, 124] jest szczególnym przypadkiem wnioskowania probabilistycznego, w którym wykonuje się operacje arytmetyczne na zmiennych losowych. O zmiennych zwykle zakłada się, że są niezależne. Z praktycznego punktu widzenia arytmetyka probabilistyczna jest probabilistyczną metodą rozwiązywania zagadnienia propagacji niepewności. Zagadnienie to stanowi zasadniczą część monografii i zostanie dokładnie omówione w rozdz. 2 (przypadek zmiennych niezależnych) oraz rozdz. 3 (przypadek zmiennych zależnych).

1.5. Istniejące biblioteki i oprogramowanie do obliczeń na zmiennych niepewnych

Wikipedia pod hasłem „Uncertainty propagation software”⁵ wyświetla listę i krótki opis około 20 różnych programów służących do obliczeń na zmiennych niepewnych. Lista ta oczywiście nie jest kompletna. Poniżej przedstawiono ważniejsze pozycje wraz z krótkim opisem:

INTLAB – pakiet obliczeń na zmiennych interwałowych dla programu Matlab, przeznaczony do obliczeń numerycznych z kontrolą dokładności [105].

b4m – kolejny pakiet obliczeń na zmiennych interwałowych dla programu Matlab [128].

Biblioteka arytmetyki interwałowej firmy Sun – podobnie jak poprzednie biblioteka przeznaczona do kontroli dokładności obliczeń numerycznych w języku C/C++ [115].

⁴ <http://www-bisc.cs.berkeley.edu/bisc>, dostęp 1.06.2012 r.

⁵ http://en.wikipedia.org/wiki/Uncertainty_propagation_software, dostęp 1.06.2012 r.

uncertainties – pakiet języka Python⁶ zawierający implementacje metody propagacji błędów [72].

Statool – pakiet umożliwiający obliczenia na zmiennych niezależnych, zmiennych zależnych o zadanym współczynniku korelacji, a także w przypadku nieznannej zależności między zmiennymi (wtedy wynikiem są dwie skrajne dystrybuanty: dolna i górna, ograniczające możliwe dystrybuanty wyniku): Program, dostarczany jako pojedyncza aplikacja, nie ma wygodnego interfejsu programistycznego (API). Dokładność obliczeń jest ograniczona ze względu na dyskretny sposób reprezentacji dystrybuant. Brak jest także możliwości precyzyjnej reprezentacji bardziej złożonych zależności między zmiennymi [8, 129].

Riskcalc – pakiet do arytmetyki probabilistycznej oparty na reprezentacji dyskretnej; obok obliczeń probabilistycznych zawiera również obliczenia rozmyte i interwałowe. Podobnie jak inne pakiety pracuje na zdyskretyzowanych dystrybuantach, nie posługuje się bezpośrednio modelami ciągłymi [29, 30].

Fuzzy Logic Toolbox – pakiet programu Matlab⁷ implementujący logikę rozmytą oraz obejmujący modele i sterowanie rozmyte. Jeśli chodzi o arytmetykę rozmytą, istnieje funkcja `fuzarith`, zaimplementowane są cztery działania dla zmiennych rozmytych, biblioteka jednak słabo się skaluje i przy bardziej złożonych obliczeniach obserwuje się znaczny spadek dokładności w trakcie wykonywania kolejnych obliczeń.

DSI Toolbox – pakiet programu Matlab implementujący aparat obliczeniowy teorii Dempstera-Shafera⁸.

Oprócz pakietów arytmetyki probabilistycznej większość implementacji opiera się na dyskretyzacji rozkładów prawdopodobieństwa, rachunku propagacji błędów lub metodzie Monte Carlo. Dostępne pakiety w większości nie umożliwiają precyzyjnego określania zależności między zmiennymi, chociaż czasami, jak w pakiecie **Statool**, mamy możliwość podania współczynnika korelacji. Warto zaznaczyć, że mając ustalone jedynie rozkłady brzegowe oraz współczynnik korelacji, rozkładu łącznego pomiędzy zmiennymi nie można wyznaczyć w sposób jednoznaczny [89, rozdz. 5]. Istnieje też bardziej subtelny problem z dzieleniem zmiennych losowych. Wynik dzielenia zmiennych losowych jest poprawnie zdefiniowany, nawet jeżeli dzielnik zawiera w nośniku zero. Ograniczeniem jest jedynie, aby zero nie było atomem miary (to znaczy, aby prawdopodobieństwo wystąpienia takiego zdarzenia było równe zero; warunek ten jest automatycznie spełniony dla rozkładów ciągłych). Na uwagę zasługuje fakt, że żaden z przebadanych pakietów obliczeniowych nie wykonywał poprawnie tego działania. **Statool** zgłaszał błąd dzielenia przez zero. Podobny problem występuje w arytmetyce interwałowej, w takich przypadkach uzyskuje się w wyniku przedziały nieskończone.

⁶ <http://packages.python.org/uncertainties>, dostęp 1.06.2012 r.

⁷ <http://www.mathworks.com>, dostęp 1.06.2012 r.

⁸ <http://www.scg.inf.uni-due.de/forschung/software/dsi-toolbox.php>, dostęp 1.06.2012 r.

Biblioteki te nie zawsze poprawnie rozróżniają przypadki niepewności identycznych – opisanych tą samą zmienną oraz opisanych taką samą niepewnością (na przykład interwałem czy rozkładem). Warto zauważyć, że rozróżnienie nie zawsze jest stosowane, co prowadzi do błędów, na przykład przy obliczaniu wyrażeń postaci $x - x$, które powinny zawsze dawać w wyniku ostrą wartość równą zero. Dokładniejsze pod tym względem są pakiety arytmetyki interwałowej [72,105]. W pracy [85] można znaleźć szczegółowy sposób postępowania w takich przypadkach.

1.6. Wnioskowanie

Często oprócz prostej propagacji niepewności jesteśmy zainteresowani bardziej złożonym wnioskowaniem. Rozważmy następujący przykład dotyczący sztafety biegaczy 4×100 m. Załóżmy, że są znane rozkłady przebiegu dystansu 100 m dla poszczególnych biegaczy oraz przyjmijmy, że przebiegi te są niezależne. Jeżeli interesuje nas rozkład sumarycznego czasu sztafety, mamy do czynienia z prostą propagacją niepewności. Natomiast nawet w takim prostym modelu możemy rozważyć znacznie bardziej skomplikowane zadania, na przykład: jakie jest prawdopodobieństwo, że drugi biegacz przebiegł swój dystans w czasie krótszym niż 9.5 s, pod warunkiem że łączny czas sztafety na mecie wyniósł 40 s. Tak postawionego zadania nie rozwiązujemy za pomocą propagacji niepewności. Podstawowy problem w tym miejscu polega na tym, że zmienne, które przed warunkowaniem były niezależne, w momencie ustalenia zmiennej wynikowej stają się zależne. Skutkuje to tym, że nie można po prostu przenieść niewiadomych na drugą stronę, tak jak postępujemy w przypadku zwykłych obliczeń liczbowych. Używając metod rachunku prawdopodobieństwa, jesteśmy w stanie kontrolować powstałe w ten sposób zależności. Jednak nawet w tak prostym przykładzie model znacznie się komplikuje, ponieważ mamy do czynienia z czterowymiarowym łącznym rozkładem zmiennych zależnych. Wyznaczenie rozkładu brzegowego jednej ze zmiennych wymaga w tym przypadku całkowania funkcji czterech zmiennych. Żadna z metod propagacji niepewności opisanych w poprzednim podrozdziale nie może być zastosowana do dokładnego rozwiązania tego zadania.

Prostsze modele zależności, zawierające małą liczbę zmiennych, mogą być opisane łącznym rozkładem danym funkcją gęstości lub za pomocą funkcji kopułowych [89]. W przypadku większej liczby zmiennych konieczne jest obniżenie wymiarowości problemu przez faktoryzację rozkładu łącznego. Ogólną metodą opisu wielowymiarowych modeli probabilistycznych pozwalającą na tego typu wnioskowanie jest koncepcja modeli graficznych (takich jak sieci bayesowskie czy sieci Markowa) [97]. Modele graficzne służą do reprezentacji wielowymiarowych modeli probabilistycznych z użyciem niskowymiarowych rozkładów warunkowych. Obecnie stosowane metody wnioskowania (czyli obliczania prawdopodobieństw brzegowych wybranych zmiennych modelu) w modelach graficznych umożliwiają uzyskanie dokładnych wyników jedynie dla zmiennych dyskretnych oraz zmiennych

ciągłych o rozkładzie normalnym [53,96]. W pozostałych przypadkach są stosowane podejścia przybliżone oparte na próbkowaniu [2,78]. Biblioteka **PaCAL** w swoich założeniach ma umożliwiać uzyskanie dokładnych wyników również w takich przypadkach, wykorzystując numeryczne reprezentacje również wielowymiarowych zmiennych losowych. Jednak opis metodologii związanej z ogólnym wnioskowaniem w modelach graficznych wykracza poza zakres monografii. Pewne typy obliczeń na zmiennych zależnych zostaną przedstawione w rozdz. 3.

2. Operacje na zmiennych losowych niezależnych

2.1. Wstęp

W poniższym rozdziale zostaną przedstawione główne wyniki dotyczące obliczeń na zmiennych losowych niezależnych. Podrozdziały 2.2 i 2.3 zawierają przypomnienie informacji z zakresu rachunku prawdopodobieństwa. W kolejnych przedstawiono szczegóły implementacyjne oraz numeryczne własności implementacji. W tej części zakładamy, że rozważane zmienne losowe mają funkcje gęstości. Twierdzenie o rozkładzie miary mówi, że dowolną miarę probabilistyczną można rozbić na trzy części: część absolutnie ciągłą względem miary Lebesgue'a, część dyskretną i część osobliwą [42, 50, 71]. Zasadnicza część pracy dotyczy przypadku zmiennych ciągłych¹ – absolutnie ciągłych względem miary Lebesgue'a w $\mathbb{R}^n$. Przypadek zmiennych ciągłych jest trudniejszy, jeżeli chodzi o implementację numeryczną i tej części zostanie poświęcone najwięcej uwagi. W dalszej części jest również rozważany przypadek zmiennych losowych dyskretnych skończonych (przyjmujących skończoną liczbę wartości) oraz przypadek zmiennych mieszanych (składających się z części absolutnie ciągłej i dyskretniej). Sam przypadek zmiennych dyskretnych skończonych jest względnie prosty w implementacji. Miary osobliwe [71], jak na przykład rozkład Cantora, nie mają skończonych reprezentacji. Z tego powodu nie są rozważane w pracy (choć niektóre z nich można aproksymować za pomocą zmiennych ciągłych lub dyskretnych [28, podrozdz. V.4]).

Zasadnicza trudność przy wykonywaniu większej liczby operacji probabilistycznych polega na zastosowaniu odpowiednich metod całkowania oraz interpolowaniu wyników pośrednich. Przedstawiona w dalszej części implementacja pakietu **PaCAL** wykorzystuje własności funkcji kawałkami gładkich oraz odpowiednio regularnych w otoczeniu punktów osobliwych i w nieskończoności. Ograniczając się do tego typu funkcji gęstości, można utrzymywać wysoką jakość aproksymacji, bliską precyzji maszynowej. Rozbicie funkcji gęstości na gładkie segmenty jest konieczne dla uzyskania odpowiedniej jakości procedur całkowania i aproksymacji. Stosowne twierdzenia pokazują, że tak zdefiniowana klasa roz-

¹ W teorii rozróżnia się przypadki zmiennych ciągłych (mających ciągłą dystrybuantę) od zmiennych absolutnie ciągłych; zmienne absolutnie ciągłe mają funkcję gęstości, która jednak nie musi być ciągła. W niniejszej pracy wyrażenie „zmienne ciągłe” jest używane w znaczeniu absolutnej ciągłości – przypadek miar osobliwych jest pominięty, szczegółowe definicje można znaleźć w pracy [71].

kładów jest zamknięta ze względu na operacje arytmetyczne i potęgowanie oraz możliwa jest ich numeryczna aproksymacja. Należy zaznaczyć też, że ograniczenie się do funkcji kawałkami gładkich odpowiednio regularnych nie jest zbyt mocne w praktyce i pokrywa praktycznie wszystkie stosowane rozkłady.

W dalszej części będą używane następujące oznaczenia i konwencje: zmienne losowe będą oznaczane wielkimi literami $X, Y, \dots$, funkcje gęstości zmiennej losowej zwykle będą oznaczane małymi literami $f, g, \dots$, a odpowiadające im dystrybuanty – wielkimi literami $F, G, \dots$ (zdefiniowane jako całki skumulowane $F(x) = \int_{-\infty}^x f(\tau) d\tau$). Zmienne losowe znanego typu będą oznaczane pochylonym fontem bezszeryfowym, na przykład: $N(0, 1)$ – zmienna o rozkładzie normalnym standaryzowanym (podrozdz. 5.3.2). Niewskazane jawnie funkcja gęstości i dystrybuanta zmiennej losowej X oznaczane będą symbolem PDF_X (ang. *Probability Density Function*) oraz CDF_X (ang. *Cumulative Density Function*). Fakt, że zmienna losowa X ma rozkład f i dystrybuantę F , oznaczono przez $X \sim f, F$. Relacja $\sim$ jest używana również do oznaczenia faktu, że X i Y mają taki sam rozkład, na przykład $X \sim Y$ lub $X \sim N(\mu, \sigma)$. Relacja $=$ jest używana w przypadku, gdy zmienne losowe są identyczne co do realizacji ($X = Y$). W szczególności jeżeli $X = Y$, to $X + Y = 2X$, natomiast jeżeli $X \sim Y$, to bez podania rozkładu łącznego tych zmiennych nie można wyznaczyć dokładnie rozkładu zmiennej $X + Y$. Takie rozróżnienie rzutuje na semantykę obliczeń (podrozdz. 5.4). W przypadku zmiennych niezależnych semantykę tę można nieco uprościć, posługując się relacją $\sim$. Jednak w ogólnym przypadku zmiennych zależnych sama relacja $\sim$ jest mniej przydatna, gdyż definiuje jedynie rozkład brzegowy, wtedy trzeba jawnie posługiwać się rozkładem łącznym (rozdz. 3).

2.2. Operacje arytmetyczne na zmiennych losowych

Podstawową monografią z zakresu obliczeń na zmiennych losowych niezależnych jest praca [113]. Opisuje ona szereg metod (głównie teoretycznych) wyznaczania rozkładów powstałych w wyniku operacji arytmetycznych na zmiennych losowych. Praca ta zawiera też dużą liczbę przykładów i ćwiczeń, które stanowią wygodny zbiór do testowania własnych metod. Podstawy teoretyczne wykonywania operacji arytmetycznych na zmiennych losowych są znane od dość dawna (na przykład książka w języku polskim M. Fisza [33] – wydanie z 1954 roku oraz kolejne). Podstawowym narzędziem teoretycznym wykorzystywanym w obliczeniach probabilistycznych jest twierdzenie o zamianie zmiennych pod znakiem całki (lub twierdzenie o całkowaniu przez podstawianie) [43, 111]. Twierdzenie to jest wykorzystywane przy wyprowadzeniu wzorów na rozkład sumy, różnicy, iloczynu, ilorazu zmiennych losowych (2.1)–(2.4), do wyznaczania funkcji zmiennych losowych, jak również w bardziej złożonych przypadkach (rozdz. 2). Użycie jakobianu przy zamianie zmiennych pod znakiem całki jest znane od czasów K.F. Gaussa (por. uwagę [124, s. 11]).

Niech X, Y będą zmiennymi losowymi opisanymi łączną funkcją gęstości h , a f, g

będą, odpowiednio, brzegowymi funkcjami gęstości. Rozkłady zmiennych losowych: $X+Y$, $X-Y$, XY , X/Y są dane całkami:

$$PDF_{X+Y}(z) = \int_{x+y=z} h(x, y) \, dx \, dy = \int_{-\infty}^{+\infty} h(x, z-x) \, dx,$$

$$PDF_{X-Y}(z) = \int_{x-y=z} h(x, y) \, dx \, dy = \int_{-\infty}^{+\infty} h(x, x-z) \, dx,$$

$$PDF_{XY}(z) = \int_{xy=z} h(x, y) \, dx \, dy = \int_{-\infty}^{+\infty} h(x, z/x) \frac{1}{|x|} \, dx,$$

$$PDF_{X/Y}(z) = \int_{x/y=z} h(x, y) \, dx \, dy = \int_{-\infty}^{+\infty} h(yz, y) |y| \, dy \\ = \int_{-\infty}^{+\infty} h(x, x/z) \frac{|x|}{z^2} \, dy.$$

Jeżeli dodatkowo założymy, że X i Y są niezależne, to jest $h(x, y) = f(x)g(y)$, wtedy wzory przybiorą postać:

$$PDF_{X+Y}(z) = \int_{-\infty}^{+\infty} f(x)g(z-x) \, dx, \quad (2.1)$$

$$PDF_{X-Y}(z) = \int_{-\infty}^{+\infty} f(x)g(x-z) \, dx, \quad (2.2)$$

$$PDF_{XY}(z) = \int_{-\infty}^{+\infty} f(x)g(z/x) \frac{1}{|x|} \, dx, \quad (2.3)$$

$$PDF_{X/Y}(z) = \int_{-\infty}^{+\infty} f(yz)g(y)|y| \, dy = \int_{-\infty}^{+\infty} f(x)g(x/z) \frac{|x|}{z^2} \, dx. \quad (2.4)$$

2.3. Funkcje zmiennych losowych

Niech u będzie funkcją ściśle monotoniczną oraz różniczkowalną, u^{-1} oznacza funkcję odwrotną, a $(u^{-1})'$ pochodną funkcji odwrotnej oraz niech $X \sim f$. Funkcja gęstości zmiennej losowej $Y = u(X)$ jest dana wzorem:

$$PDF_{u(X)}(x) = g(u^{-1}(x)) \cdot |(u^{-1})'(x)|. \quad (2.5)$$

W szczególności funkcja gęstości odwrotności zmiennej losowej $X \sim f$ jest wyrażona zależnością:

$$PDF_{1/X}(x) = f(1/x)/x^2.$$

Ważnymi operacjami, często stosowanymi w praktyce, są przesunięcie i skalowanie zmiennej losowej (czyli transformacja afiniczna). Rozkład zmiennej $aX + b$ wyraża się wzorem:

$$PDF_{aX+b}(x) = \frac{1}{a} f\left(\frac{x-b}{a}\right).$$

Jeżeli funkcja u nie jest monotoniczna, ale składa się ze skończonej liczby segmentów, które są różniczkowalne i ściśle monotoniczne, to rozkład zmiennej losowej $u(X)$ daje się przedstawić w postaci sumy:

$$PDF_{u(X)}(x) = \sum_i g(u_i^{-1}(x)) \cdot |(u_i^{-1})'(x)|. \quad (2.6)$$

W szczególności kwadrat zmiennej losowej $X \sim f$ ma funkcję gęstości daną wzorem:

$$PDF_{X^2}(x) = \frac{1}{2\sqrt{x}} (f(-\sqrt{x}) + f(\sqrt{x})).$$

Przykład 2.1. Niech X będzie zmienną losową o rozkładzie χ_1^2 (chi-kwadrat z jednym stopniem swobody). Funkcja gęstości zmiennej X wyraża się wzorem:

$$f(x) = \frac{1}{\sqrt{2}\Gamma(\frac{1}{2})} x^{-\frac{1}{2}} \exp\left(-\frac{x}{2}\right).$$

Rozważmy rozkład zmiennej $1/X$. Na podstawie równania (2.5) ma on postać:

$$g(x) = f(1/x)/x^2 = \frac{1}{\sqrt{2}\Gamma(\frac{1}{2})} x^{-\frac{1}{2}} \exp\left(-\frac{1}{2x}\right)/x^2 = \frac{1}{\sqrt{2}\Gamma(\frac{1}{2})} x^{-\frac{3}{2}} \exp\left(-\frac{1}{2x}\right).$$

Jest to funkcja gęstości rozkładu Levy'ego [28]. Ponieważ $\exp(-1/(2x)) \rightarrow 1$ dla $x \rightarrow \infty$, więc $g(x) = \mathcal{O}(x^{-1.5})$ dla $x \rightarrow \infty$. Wynika stąd, że rozkład ten nie może mieć ani średniej, ani momentów wyższego rzędu. Natomiast rozkład χ^2 ma oczywiście momenty wszystkich rzędów.

Przykład ten pokazuje, że dzielenie nie jest operacją dobrze określoną w terminach momentów zmiennej losowej. Ponadto pośrednio wskazuje to na ograniczenia w posługiwaniu się definicją momentów.

2.4. Operacje arytmetyczne z użyciem funkcji kawałkami gładkich

2.4.1. Funkcje kawałkami gładkie

O funkcji gęstości zakładamy jedynie, że jest to funkcja dodatnio określona i unormowana (w normie $\|\cdot\|_1$) do jedności. W ogólności taka funkcja może mieć szereg właściwości powodujących trudności analityczne. Na przykład może zawierać punkty osobliwe (w których funkcja lub jej pochodne nie są określone lub w otoczeniu których zmierzają do nieskończoności), może być określona na przedziale niewłaściwym, jak na przykład funkcja gęstości rozkładu χ^2 z jednym stopniem swobody (porównaj przykład 2.1). Takie funkcje wymagają osobnych metod interpolacji oraz całkowania. Pewne rozkłady, jak na przykład rozkład jednostajny, mimo że są regularne na przedziale, na którym są określone, to jed-

nak nie są regularne na przedziale $(-\infty, +\infty)$. Natomiast takiego przedziału całkowania wymagamy, jeżeli drugi ze składników sumy jest określony na przedziale nieskończonym. Aby zapanować nad sytuacją z numerycznego punktu widzenia, wprowadzone zostaną funkcje kawałkami gładkie, które są podstawą reprezentacji numerycznej.

Mówimy, że funkcja jest kawałkami gładka, jeżeli jej dziedzinę da się rozbić na skończoną liczbę rozłącznych przedziałów w taki sposób, że odciecie tej funkcji do poszczególnych przedziałów jest gładkie. Funkcję kawałkami gładką można przedstawić jako skończoną sumę gładkich segmentów:

$$f(x) = \sum_i f_i(x) \chi_{[a_i, b_i]}(x),$$

gdzie $f^i \in C^k[a_i, b_i]$ oraz $\chi_{[a_i, b_i]}$ jest funkcją charakterystyczną przedziału $[a_i, b_i]$:

$$\chi_{[a_i, b_i]}(x) = \begin{cases} 1, & x \in [a_i, b_i], \\ 0 & \text{dla pozostałych.} \end{cases}$$

Niech X i Y będą niezależnymi zmiennymi losowymi, a $f = \sum_{i=1}^n f_i(x) \chi_{[a_i, b_i]}(x)$ i $g = \sum_{j=1}^m g_j(x) \chi_{[a_j, b_j]}(x)$ będą kawałkami gładkimi funkcjami gęstości tych zmiennych. Na podstawie wzorów (2.1), (2.2), (2.3) i (2.4) można wyprowadzić analogiczne wzory dla funkcji kawałkami gładkich.

2.4.2. Dodawanie i odejmowanie

Funkcja gęstości sumy zmiennych losowych $X + Y$ może być przedstawiona w postaci:

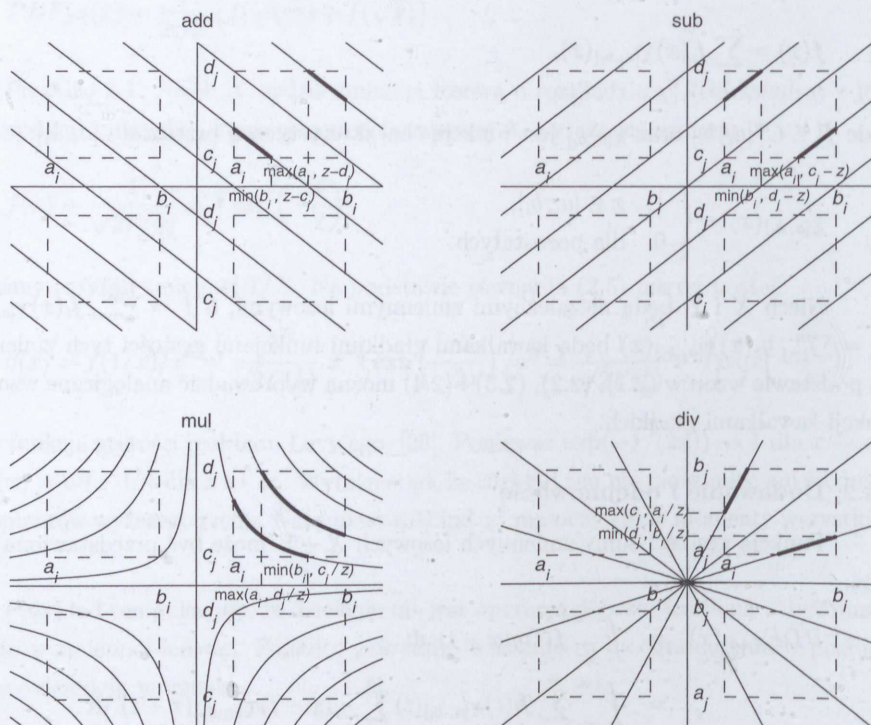
$$\begin{aligned} PDF_{X+Y}(x) &= \int_{-\infty}^{+\infty} f(t)g(x-t) dt \\ &= \int_{-\infty}^{+\infty} \sum_{i=1}^n f_i(t) \chi_{[a_i, b_i]}(t) \sum_{j=1}^m g_j(x-t) \chi_{[c_j, d_j]}(x-t) dt \\ &= \sum_{i=1}^n \sum_{j=1}^m \int_{-\infty}^{+\infty} f_i(t) \chi_{[a_i, b_i]}(t) g_j(x-t) \chi_{[c_j, d_j]}(x-t) dt \\ &= \sum_{i=1}^n \sum_{j=1}^m \int_{\max(a_i, x-d_j)}^{\min(b_i, x-c_j)} f_i(t) g_j(x-t) dt. \end{aligned} \quad (2.7)$$

Wynik, to znaczy rozkład zmiennej $X + Y$, jest również funkcją kawałkami gładką, jeśli funkcje f i g są kawałkami gładkie. Wynika to z własności całki i splotu [35, 44]. Punkty podziału segmentów rozkładu wynikowego mają postać $\{a_i + b_j, c_i + d_j, i = 1, \dots, n, j = 1, \dots, m\}$.

Idea całkowania wzdłuż odpowiednich konturów dla czterech podstawowych operacji arytmetycznych jest przedstawiona na rys. 2.1. Całkowanie jest dobrze określone

jedynie w przypadku, gdy $\max(a_i, x - d_i) < \min(b_i, x - c_i)$, co jest równoważne $a_i + c_j < x < b_i + d_j$. Warunki te wyznaczają granice dziedziny segmentów rozkładu wynikowego. W pozostałych przypadkach całki w powyższej sumie mają wartość zero. Geometrycznie odpowiada to przypadkowi, w którym kontur całkowania ($x + y = z$) nie przecina się z iloczynem kartezjańskim przedziałów $[a_i, b_i] \times [c_j, d_j]$.

W przypadku odejmowania wszystkie obliczenia, z dokładnością do zmiany znaku, przebiegają analogicznie.



Rys. 2.1. Kontury całkowania dla czterech podstawowych operacji, z zaznaczonymi granicami całkowania wyznaczonymi przez segmenty

2.4.3. Mnożenie

Funkcja gęstości iloczynu zmiennych losowych XY może być przedstawiona w postaci:

$$\begin{aligned}
 PDF_{XY}(x) &= \int_{-\infty}^{+\infty} f(t)g(x/t) \frac{1}{|t|} dt \\
 &= \int_{-\infty}^{+\infty} \sum_{i=1}^n f_i(t) \chi_{[a_i, b_i]}(t) \sum_{j=1}^m g_j(x/t) \chi_{[c_j, d_j]}(x/t) \frac{1}{|t|} dt \\
 &= \sum_{i=1}^n \sum_{j=1}^m \int_{-\infty}^{+\infty} f_i(t) \chi_{[a_i, b_i]}(t) g_j(x/t) \chi_{[c_j, d_j]}(x/t) \frac{1}{|t|} dt \\
 &= \begin{cases} \sum_{i=1}^n \sum_{j=1}^m \int_{\max(a_i, x/d_j)}^{\min(b_i, x/c_j)} f_i(t) g_j(x/t) \frac{1}{|t|} dt, & x > 0, \\ \sum_{i=1}^n \sum_{j=1}^m \int_{a_i}^{b_i} f_i(t) g_j(0^+) \frac{1}{|t|} dt, & x = 0^+, \\ \sum_{i=1}^n \sum_{j=1}^m \int_{a_i}^{b_i} f_i(t) g_j(0^-) \frac{1}{|t|} dt, & x = 0^-, \\ \sum_{i=1}^n \sum_{j=1}^m \int_{\max(a_i, x/c_j)}^{\min(b_i, x/d_j)} f_i(t) g_j(x/t) \frac{1}{|t|} dt, & x < 0. \end{cases} \quad (2.8)
 \end{aligned}$$

Analogiczne całkowanie można przeprowadzić wzdłuż drugiej zmiennej :

$$PDF_{XY}(x) = \begin{cases} \sum_{i=1}^n \sum_{j=1}^m \int_{\max(c_j, x/b_i)}^{\min(d_j, x/a_i)} f_i(x/t) g_j(t) \frac{1}{|t|} dt, & x > 0, \\ \sum_{i=1}^n \sum_{j=1}^m \int_{c_j}^{d_j} f_i(0^+) g_j(t) \frac{1}{|t|} dt, & x = 0^+, \\ \sum_{i=1}^n \sum_{j=1}^m \int_{c_j}^{d_j} f_i(0^-) g_j(t) \frac{1}{|t|} dt, & x = 0^-, \\ \sum_{i=1}^n \sum_{j=1}^m \int_{\max(c_j, x/a_i)}^{\min(d_j, x/b_i)} f_i(x/t) g_j(t) \frac{1}{|t|} dt, & x < 0. \end{cases} \quad (2.9)$$

Podobnie jak w przypadku sumy rozkład zmiennej XY jest funkcją kawałkami gładką oraz węzły podziału segmentów są unikalnymi punktami ze zbioru $\{a_i c_j, a_i d_j, b_i c_j, b_i d_j, i = 1, \dots, n, j = 1, \dots, m\}$. Całkowanie ma sens, jeżeli kontur całkowania przecina się z iloczynem kartezjańskim dziedzin segmentów $[a_i, b_i] \times [c_j, d_j]$ (rys. 2.1). Warto zauważyć, że osobnego potraktowania wymaga poziomica $xy = 0$, której rozwiązaniem są osie układu współrzędnych. Rozróżnienie pomiędzy 0^+ i 0^- jest konieczne w przypadku, gdy jeden z czynników ma nieciągłość w punkcie 0. Jest to dość częsta sytuacja, na przykład nieciągły w zerze jest rozkład jednostajny na odcinku $[0, 1]$ czy rozkład wykładniczy. Ponadto zero jest punktem osobliwym jakobianu, co powoduje, że przedział całkowania w otoczeniu zera i tak trzeba rozbić na dwie części: dodatnią i ujemną². Z tego powodu wygodnie jest zapewnić, aby zero było jednym z punktów podziału segmentów dla obu czynników. Widać również, że mnożenie rozkładów zawierających w nośniku zero powoduje trudności numeryczne ze względu na konieczność liczenia całek niewłaściwych.

² Należy tak zrobić, ponieważ nie spotyka się procedur całkowania funkcji mających punkty osobliwe wewnątrz przedziału całkowania, zwykle integratory numeryczne dopuszczają osobliwości na końcach przedziałów całkowania (patrz na przykład plik pomocy do funkcji `quadgk` programu Matlab, www.mathworks.com, dostęp 1.06.2012 r.).

2.4.4. Dzielenie

Funkcja gęstości ilorazu zmiennych losowych X/Y jest przedstawiona w postaci:

$$\begin{aligned}
 PDF_{X/Y}(x) &= \int_{-\infty}^{+\infty} f(xt)g(t)|t| \, dt \\
 &= \int_{-\infty}^{+\infty} \sum_{i=1}^n f_i(xt)\chi_{[a_i, b_i]}(xt) \sum_{j=1}^m g_j(x)\chi_{[c_j, d_j]}(x)|t| \, dt \\
 &= \sum_{i=1}^n \sum_{j=1}^m \int_{-\infty}^{+\infty} f_i(xt)\chi_{[a_i, b_i]}(xt)g_j(x)\chi_{[c_j, d_j]}(x)|t| \, dt \\
 &= \sum_{i=1}^n \sum_{j=1}^m \begin{cases} \int_{\max(c_j, a_i/x)}^{\min(d_j, b_i/x)} f_i(xt)g_j(t)|t| \, dt, & x > 0, \\ \int_{c_j}^{d_j} f_i(t0^+)g_j(t)|t| \, dt, & x = 0^+, \\ \int_{c_j}^{d_j} f_i(t0^-)g_j(t)|t| \, dt, & x = 0^-, \\ \int_{\max(c_j, b_i/x)}^{\min(d_j, a_i/x)} f_i(xt)g_j(t)|t| \, dt, & x < 0. \end{cases} \quad (2.10)
 \end{aligned}$$

Analogicznie całkując wzdłuż pierwszej zmiennej:

$$PDF_{X/Y}(x) = \begin{cases} \sum_{i=1}^n \sum_{j=1}^m \int_{\max(a_i, c_j x)}^{\min(b_i, d_j x)} f_i(t)g_j(t/x) \frac{|t|}{x^2} \, dt, & x > 0, \\ \sum_{i=1}^n \sum_{j=1}^m \int_{\max(a_i, d_j x)}^{\min(b_i, c_j x)} f_i(t)g_j(t/x) \frac{|t|}{x^2} \, dt, & x < 0. \end{cases} \quad (2.11)$$

Rozkład zmiennej X/Y jest funkcją kawałkami gładką. Węzłami podziału są unikalne punkty ze zbioru $\{a_i/c_j, a_i/d_j, b_i/c_j, b_i/d_j, i = 1, \dots, n, j = 1, \dots, m\}$. Całkowanie, podobnie jak w przypadku mnożenia, ma sens, jeżeli poziomicą $x/y = z$ ma niepuste przecięcie z iloczynem kartezjańskim przedziałów $[a_i, b_i] \times [c_j, d_j]$ (rys. 2.1).

Warto zauważyć, że poprzednie wzory na sumę i iloczyn są w pewnym sensie symetryczne (nie zależą od tego, którą ze zmiennych wybierzemy do podstawienia, jeżeli rozkłady f i g są identyczne). W obu przypadkach całki, które należy policzyć, mają podobną postać. Jednak dla ilorazu całki te wyglądają inaczej, mimo że ich wynikiem jest ta sama funkcja gęstości. Na przykład całka (2.10) jest dobrze określona dla $x = 0$, natomiast druga z całek (2.11) nie (zakładając, że f i g są ciągłe w punkcie 0). Może to powodować trudności przy interpolacji, jeżeli jeden z węzłów interpolacji jest ustawiony w punkcie 0. Ponadto w obu przypadkach mamy do scałkowania dwie różne funkcje. Wybór odpowiedniej z nich może znacznie ułatwić proces całkowania.

Wzory (2.8), (2.9) i (2.10) oraz kontury całkowania przedstawione na rys. 2.1 pokazują, że zero jest wyróżnione w pewien sposób. Z tego powodu wygodnie jest przyjąć, że w zerze jest również granica segmentów zarówno argumentów, jak i wyniku, mimo że osobliwość w zerze może wystąpić lub nie. To, czy wystąpi, zależy od postaci argumentów (funkcji f i g) oraz wykonywanego działania. Widać to dobrze na przykładzie iloczynu i ilorazu dwóch niezależnych zmiennych losowych o rozkładzie normalnym [113, podrozdz. 4.4.1, podrozdz. 4.5.2]. Iloczyn w zerze ma wyraźną nieregularność (funkcja gęstości z obu stron zera dąży do nieskończoności), a iloraz jest bardzo regularnym rozkładem Cauchy'ego. Natomiast w przypadku iloczynu i ilorazu niezależnych zmiennych losowych

o rozkładzie Cauchy'ego dostajemy w wyniku identyczny rozkład³ mający osobliwość w zerze zmierzającą do nieskończoności [113, s. 158].

Zapis dzielenia zmiennych losowych w postaci:

$$X/Y = X \cdot (1/Y) \quad (2.12)$$

sugeruje, że dysponując odwrotnością, można by zaimplementować jedynie mnożenie. Jednak przeprowadzone eksperymenty numeryczne pokazują, że w wielu przypadkach bezpośrednie całkowanie rozkładu ilorazowego daje lepsze rezultaty, gdyż funkcje do całkowania są często bardziej regularne. Taka sama uwaga dotyczy posługiwania się zależnościami (2.10) oraz (2.11), dla obu tych formuł można znaleźć numeryczne przykłady ilorazów preferujące jedną z nich. W pakiecie **PaCAL** na podstawie analizy wielu przykładów numerycznych dzielenie wykonuje się bezpośrednio, używając w zależności od sytuacji wzoru (2.10) lub (2.11). Rzecz jasna, w większości prostych przypadków wyniki numeryczne pokrywają się z zależnością (2.12).

Wzory (2.7)–(2.9) pozwalają dokładnie kontrolować przedziały, w których całkowane funkcje są gładkie, oraz dokładnie wyznaczyć segmenty rozkładu wynikowego. Przedstawienia te są podstawą implementacji czterech operacji arytmetycznych na zmiennych losowych niezależnych.

2.4.5. Funkcje zmiennych losowych

Funkcje zmiennych losowych wyznacza się bezpośrednio na podstawie (2.6), o ile tylko węzły podziału dziedziny ustawimy tak, aby funkcje określone na segmentach były monotoniczne. Warto jednak zwrócić uwagę, że automatyzacja tego typu operacji jest dość złożona, ponieważ wymaga znalezienia funkcji odwrotnej oraz różniczkowania funkcji odwrotnej, a w sytuacji gdy funkcja wyjściowa nie jest monotoniczna, dodatkowo konieczne jest rozwiązywanie równań. W przypadku pewnej liczby funkcji standardowych wszystkie potrzebne informacje (jak funkcje odwrotne, pochodne czy przedziały monotoniczności) można stablicować.

2.5. Minimum i maksimum, rozkłady statystyk pozycyjnych

W podobny sposób jak operacje arytmetyczne można wyznaczyć praktycznie każdy rodzaj operacji dwuargumentowej. Niech $X \sim f, F, Y \sim g, G$ będą niezależnymi zmiennymi losowymi oraz załóżmy, że interesuje nas rozkład zmiennych losowych $\min(X, Y)$

³ Dzieje się tak dlatego, że rozkład Cauchy'ego jest rozkładem ilorazowym, będącym wynikiem ilorazu niezależnych zmiennych losowych o takim samym rozkładzie (ang. *ratio distribution*). Dla niezależnych zmiennych o jednakowym rozkładzie ilorazowym iloczyn i iloraz mają identyczny rozkład.

oraz $\max(X, Y)$. Rozkłady tych zmiennych wygodniej jest napisać w terminach dystrybuant niż funkcji gęstości, gdyż mamy:

$$\begin{aligned} CDF_{\min(X,Y)}(z) &= \Pr(\min(X, Y) < z) = \Pr(X < z \wedge Y < z) \\ &= \Pr(X < z) \Pr(Y < z) = CDF_X(z) CDF_Y(z). \end{aligned}$$

Dystrybuantę maksimum zmiennych losowych uzyskuje się analogicznie, dostając ostatecznie:

$$CDF_{\min(X,Y)}(z) = F(z)G(z), \quad (2.13)$$

$$CDF_{\max(X,Y)}(z) = (1 - F(z))(1 - G(z)). \quad (2.14)$$

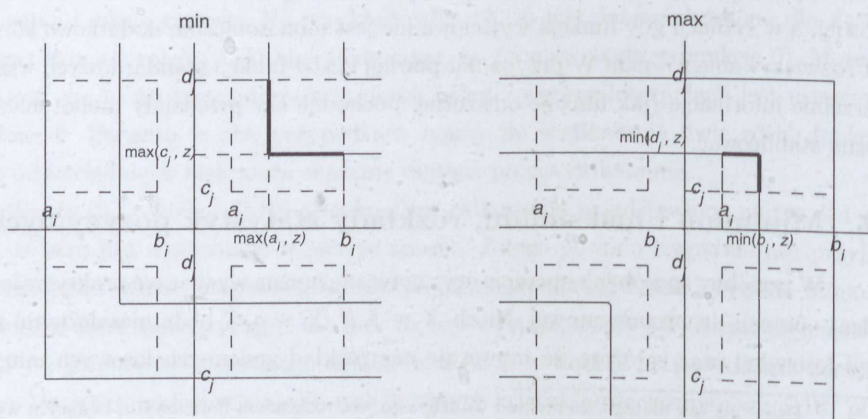
Różniczkowanie obu wyrażeń oraz podstawienie w miejsce dystrybuanty skumulowanej całki z funkcji gęstości pozwala wyznaczyć gęstość obu zmiennych losowych jedynie w terminach funkcji gęstości argumentów:

$$PDF_{\min(X,Y)}(z) = f(z) \int_{-\infty}^z g(y) dy + g(z) \int_{-\infty}^z f(x) dx, \quad (2.15)$$

$$PDF_{\max(X,Y)}(z) = f(z) \int_z^{+\infty} g(y) dy + g(z) \int_z^{+\infty} f(x) dx. \quad (2.16)$$

Całkowania występujące we wzorach (2.15) i (2.16) odpowiadają całkowaniu wzdłuż konturów $\min(x, y) = z$ oraz $\max(x, y) = z$, analogicznie jak w przypadku operacji arytmetycznych. Ilustrację graficzną całkowania łącznej funkcji gęstości wewnątrz segmentów przedstawiono na rys. 2.2.

Częstym modelem wnioskowania statystycznego są próby pobrane z ustalonej populacji w sposób niezależny, czyli zbiory niezależnych zmiennych losowych o jednakowym



Rys. 2.2. Kontury całkowania dla operacji minimum i maksimum, z zaznaczonymi granicami całkowania wyznaczonymi przez segmenty

rozkładzie (ang. *i.i.d. independent and identically distributed*). W takim modelu powyższe wzory można łatwo rozszerzyć na dowolne statystyki pozycyjne z próby, a minimum i maksimum są w takim przypadku skrajnymi statystykami pozycyjnymi.

Niech $X_i \sim f, F$ dla $i = 1, \dots, n$ będzie zbiorem niezależnych zmiennych losowych, a $X_{(i)}$ oznacza i -tą statystykę pozycyjną zbioru $\{X_1, \dots, X_n\}$ [42, podrozdz. 2.23]. Dystrybuanta oraz funkcja gęstości i -tej statystyki pozycyjnej dane są wzorami [110]:

$$\begin{aligned} CDF_{X_{(i)}}(z) &= \sum_{i=k}^n \binom{n}{i} (F(z))^i (1 - F(z))^{n-i}, \\ PDF_{X_{(i)}}(z) &= \binom{n}{i-1, 1, n-i} (F(z))^{i-1} (1 - F(z))^{n-i} f(z). \end{aligned}$$

Pierwsza z tych równości jest prostą koniunkcją warunków, drugą równość uzyskujemy z pierwszej przez różniczkowanie. Wyrażenie $\binom{n}{i-1, 1, n-i}$ oznacza symbol wielomianowy określony wzorem:

$$\binom{n}{k_1, \dots, k_s} = \frac{n!}{k_1! \cdot \dots \cdot k_s!}.$$

W pewnych przypadkach interesujące jest wykonywanie działań na statystykach pozycyjnych. W tym miejscu należy jednak zauważyć, że zmienne te są zależne i do wykonywania tych operacji potrzebny jest łączny rozkład statystyk pozycyjnych. Zagadnienie to zostanie omówione w podrozdz. 3.3.4.

2.6. Zmienne dyskretne i mieszane

Zmienne dyskretne przyjmujące skończoną liczbę wartości są prostsze w implementacji niż zmienne ciągłe. Z technicznego punktu widzenia zmienne dyskretne przyjmujące skończoną liczbę wartości mogą być łatwo reprezentowane z użyciem tablic.

Niech $X \sim f$ i $Y \sim g$ będą dyskretnymi zmiennymi losowymi określonymi, odpowiednio, na zbiorach $\{x_1, \dots, x_n\}$ i $\{y_1, \dots, y_m\}$ oraz niech $f(x_i) = \Pr(X = x_i) = f_i$, $g(y_i) = \Pr(Y = y_i) = g_i$. W przypadku dyskretnym wzory (2.1)–(2.4) redukują się do zwykłych sum:

$$PDF_{X+Y}(z) = \sum_{x_i+y_j=z} f(x_i)g(y_j), \quad (2.17)$$

$$PDF_{X \cdot Y}(z) = \sum_{x_i \cdot y_j=z} f(x_i)g(y_j), \quad (2.18)$$

$$PDF_{X/Y}(z) = \sum_{x_i/y_j=z} f(x_i)g(y_j). \quad (2.19)$$

Dziedzina zmiennej $X \circ Y$, $\circ \in \{+, -, \cdot, /\}$ jest równa zbiorowi unikalnych wartości działań postaci $x_i \circ y_j$, gdzie $x_i \in \{x_1, \dots, x_n\}$, $y_j \in \{y_1, \dots, y_m\}$. Wyrażenia te sugerują, że do obliczenia rozkładów dyskretnych zmiennych $X \circ Y$ potrzeba $\mathcal{O}(n^3)$ prostych operacji zmiennopozycyjnych (przy założeniu, że $m \approx n$). Jednak wykorzystanie własności tablic haszujących [65, podrozdz. 6.4] czy słowników w języku Python powala na zredukowanie liczby prostych operacji do $\mathcal{O}(nm)$. To jest mniej więcej tyle, ile potrzeba do wyznaczenia rozkładu łącznego dwóch zmiennych dyskretnych.

Przypadek zmiennych mieszanych

Pewnym problemem przy reprezentacji funkcji gęstości jest łączenie zmiennych ciągłych i dyskretnych. Zmienne dyskretne można w pewnym stopniu ujednolicić w zapisie ze zmiennymi ciągłymi, używając w dystrybucji δ Diraca [92]. Przy liczeniu całek (2.1)–(2.4) rozkłady należy rozdzielić na część ciągłą i dyskretną oraz osobno potraktować trzy przypadki:

1. *Przypadek ciągły* – oba argumenty ciągłe, całki postaci (2.7)–(2.11) są liczone bezpośrednio.

2. *Przypadek dyskretny* – oba argumenty dyskretne, wystarcza bezpośrednio sumowanie, jak we wzorach (2.17)–(2.19).

3. *Przypadek mieszany* – jeden z argumentów jest ciągły, a drugi dyskretny, w tym przypadku wykorzystuje się własności dystrybucji δ Diraca, co w efekcie sprowadza się do sumowania odpowiednio przesuniętych funkcji ciągłych. W przypadku mieszanym występują dwa podprzypadki, w zależności od tego, który z argumentów jest ciągły. Ma to znaczenie zwłaszcza dla odejmowania i dzielenia, ponieważ operacje te nie są przemienne.

Rozpatrzmy przypadek, gdy jedna ze zmiennych jest zmienną dyskretną ($X \sim f$), a druga ciągłą ($Y \sim g$). Rozkład zmiennej dyskretnej X można zapisać w postaci:

$$f(x) = \sum_{i=1}^n f_i \delta(x - x_i).$$

W takim przypadku całka (2.1) dla sumy zmiennych losowych $X + Y$ przyjmuje postać:

$$\begin{aligned} PDF_{X+Y}(x) &= \int_{-\infty}^{\infty} f(x-t)g(t) dt = \int_{-\infty}^{\infty} \sum_{i=1}^n f_i \delta(x-x_i-t)g(t) dt \\ &= \sum_{i=1}^n f_i \int_{-\infty}^{\infty} \delta(x-x_i-t)g(t) dt = \sum_{i=1}^n f_i g(x-x_i). \end{aligned}$$

Widać, że wynik jest zmienną ciągłą. Jest to mieszanica odpowiednio poprzesuwaných rozkładów g i dodanych z wagami f_i . Analogiczne wyniki dostajemy w przypadku pozo-

stałych działań arytmetycznych:

$$PDF_{X \cdot Y}(x) = \sum_{i=1}^n f_i g(x/x_i)/|x_i|,$$

$$PDF_{X/Y}(x) = \sum_{i=1}^n f_i g(x_i/x)/x^2.$$

Podobnie jak w przypadku dodawania, wynik obu operacji jest zmienną ciągłą. Analogicznie można podać wyrażenia dla operacji minimum i maksimum, z tym zastrzeżeniem, że wyniki tych operacji w ogólności mogą zawierać zarówno część dyskretną, jak i ciągłą.

2.7. Wybór sposobu reprezentacji

2.7.1. Transformacje i momenty

Wybór sposobu reprezentacji zmiennych losowych musi być dostosowany do rodzaju wykonywanych operacji. Istnieją reprezentacje dopasowane dobrze do szczególnych rodzajów zmiennych losowych czy do operacji szczególnego typu. Takie szczególne przypadki mogą być zwykle realizowane bardzo wydajnie. Na przykład sumowanie zmiennych normalnych można wykonywać bardzo wydajnie, ograniczając się do obliczeń zmienno-pozycyjnych na średnich i wariancjach. Jednak z punktu widzenia projektu **PaCAL** bardziej interesujące są reprezentacje ogólne, obejmujące większość rozkładów prawdopodobieństwa używanych w praktyce oraz umożliwiające wykonywanie większości operacji na zmiennych losowych. Poniżej przedstawiono przegląd możliwych i spotykanych reprezentacji zmiennych losowych.

Reprezentacje oparte na transformatach różnego typu w pewnych przypadkach pozwalają wykorzystać własności tych transformat do obliczeń na zmiennych losowych. Na przykład transformata Fouriera i własności spłotu są wykorzystywane przy dodawaniu zmiennych losowych niezależnych. W takim przypadku spłot nie wymaga całkowania i jest liczony jako iloczyn transformat oraz wymaga zastosowania transformacji odwrotnej. Podobnie transformata Melina i własności spłotu Melina mogą być wykorzystywane przy wykonywaniu mnożenia [19, 113, 124]. Metody te są jednak zbyt szczególne, żeby można było je wykorzystać w każdym przypadku. Na przykład funkcja charakterystyczna (transformata Fouriera funkcji gęstości w przypadku ciągłym) rozkładu jednostajnego na odcinku $[-1, 1]$ wyraża się następująco:

$$\mathcal{F}(PDF_{U(-1,1)}) = \frac{\exp(it) - \exp(-it)}{2it} = \sin(t)/t, \quad t \in (-\infty, \infty).$$

Jak widać, jest to funkcja trudna do numerycznej aproksymacji. W przypadku pewnej liczby rozkładów tego typu transformaty można by stabilizować, jednak prędzej czy później pojawi się konieczność numerycznego wyznaczania transformat oraz transformat

odwrotnych (na przykład przy dzieleniu). Mimo że funkcja charakterystyczna jest zawsze dobrze określona, to jednak pozostaje ona z tego punktu widzenia narzędziem raczej teoretycznym niż ogólnym narzędziem numerycznym.

Transformata Laplace'a funkcji gęstości (funkcja tworząca momenty zmiennej losowej) jest również często wykorzystywana w praktyce [28, 92]. W wielu przypadkach działania na rozkładach przenoszą się na odpowiednie działania na momentach. Jest tak na przykład przy dodawaniu. Reprezentacje za pomocą momentów są również użyteczne i bardzo często wykorzystywane w praktyce [66]. Rachunek błędów częściowo opiera się na rozwinięciu względem momentów. Reprezentacje te jednak są ograniczone do przypadku, w którym takie momenty występują. Analogiczna sytuacja występuje w przypadku kumulant. Zasadniczymi zaletami tego podejścia są łatwość implementacji i szybkość obliczeń. Problemem jest w tym przypadku odtwarzanie funkcji gęstości. Jeżeli danych jest skończenie wiele momentów, odtworzenie funkcji gęstości nie jest w ogólności jednoznaczne. Odtwarzanie funkcji na podstawie momentów znane jest pod nazwą problemu momentów Hamburgera (ang. *Hamburger moment problem*) [70].

Momenty w zasadzie wymuszają interpretowanie wyników w terminach wartości średniej i odchylenia standardowego. Użycie momentów jako reprezentacji ogranicza również zakres stosowanych działań. Dzielenie dwóch zmiennych losowych, które mają momenty wszystkich rzędów, może w rezultacie dać wynik, który nie ma żadnego momentu. Najbardziej chyba znanym przykładem jest tu iloraz niezależnych zmiennych o standaryzowanym rozkładzie normalnym, który w wyniku daje rozkład Cauchy'ego [28, 113] (por. przykład 2.1).

2.7.2. Numeryczna reprezentacja funkcji gęstości i dystrybuanty

Podstawową operacją wykonywaną w trakcie obliczeń na zmiennych losowych jest całkowanie. Wyrażenia na rozkłady minimum (2.13) i maksimum (2.14) zmiennych losowych niezależnych mogą sugerować reprezentację zmiennej losowej z użyciem dystrybuanty. Takie podejście może też wydawać się nawet bardziej naturalne dla praktyka, ponieważ większość użytecznych statystyk sumarycznych zmiennej losowej uzyskujemy przez całkowanie funkcji gęstości. Używając dystrybuanty, w niektórych przypadkach można uniknąć całkowania. Jednak podstawową różnicę widać w sposobie wykonywania działań arytmetycznych. Wzory (2.1)–(2.4) można przepisać w terminach dystrybuant. Na przykład niech dystrybuanty zmiennych losowych X i Y będą równe, odpowiednio, $F(x) = \int_{-\infty}^x f(t) dt$ i $G(x) = \int_{-\infty}^x g(t) dt$. Wtedy dystrybuanta zmiennej losowej $X + Y$ wyraża się wzorem:

$$CDF_{X+Y}(z) = \Pr(X + Y < z) = \int_{-\infty}^{+\infty} F(x - t)g(t) dt. \quad (2.20)$$

Jak widać, oba wyrażenia, (2.1) oraz (2.20), wymagają całkowania. Z technicznego punktu widzenia zastosowanie dystrybuant jest jednak bardziej problematyczne, gdyż w trakcie obliczeń i tak potrzebna jest funkcja gęstości. Wymaga to dodatkowej operacji obliczania pochodnej jednego z argumentów. Obliczanie dystrybuanty w większości praktycznych przypadków jest konieczne, na przykład do wyznaczania przedziałów ufności. Jednak dystrybuantę można wyznaczyć jednym całkowaniem na koniec obliczeń, natomiast korzystając z zależności (2.20), pochodną trzeba wyznaczać przy każdej operacji. Jeżeli przyjąć, że z technicznego punktu widzenia czas obliczania pochodnej funkcji jest porównywalny do czasu obliczania całki skumulowanej (podrozdz. 4.3.3, 4.3.4), to widać, że formuła (2.20) jest nieco bardziej złożona niż (2.1), gdyż wymaga zarówno całkowania, jak i obliczania pochodnej. Niemniej zasadniczy problem w posługiwaniu się zależnością (2.20) polega na utrzymaniu wysokiej dokładności przy obliczaniu pochodnych funkcji, zwłaszcza takich, których pochodna ma osobliwości (jak na przykład arcsin na przedziale $[-1, 1]$ czy dystrybuanta rozkładu χ^2). Całkowanie, w przeciwieństwie do różniczkowania, poprawia własności analityczne funkcji, dzięki temu w praktyce całkowanie można wykonywać z wyższą dokładnością. Z tych powodów w pakiecie **PaCAL** stosuje się numeryczne reprezentacje funkcji gęstości wykorzystujące funkcje kawałkami gładkie.

2.8. Klasy rozkładów zamknięte ze względu na operacje arytmetyczne i potęgowanie

Rozwiązania numeryczne zaimplementowane w pakiecie **PaCAL** umożliwiają przeprowadzanie obliczeń na szerokiej klasie rozkładów oraz wykonywanie działań na zmiennych losowych niezależnych z dokładnością bliską precyzji obliczeń zmiennopozycyjnych. Wyniki prezentowane w tym podrozdziale w zasadniczej części (z wyjątkiem przykładu 2.3) zostały opublikowane w pracy [51].

Jak widać z rozważań z poprzedniego podrozdziału, zbiór zmiennych losowych mających momenty jest zamknięty ze względu na dodawanie, jednak nie jest zamknięty ze względu na dzielenie. Poszukując odpowiednio regularnych zbiorów zmiennych losowych, należy być świadomym pewnych ograniczeń wynikających bezpośrednio ze wzorów (2.1)–(2.4), jak i z uwarunkowań numerycznych. Rozważmy następujący przykład [51]:

Przykład 2.2 (iloraz funkcji aproksymowalnych). Niech X i Y będą niezależnymi zmiennymi losowymi o jednakowej funkcji gęstości danej za pomocą szeregu:

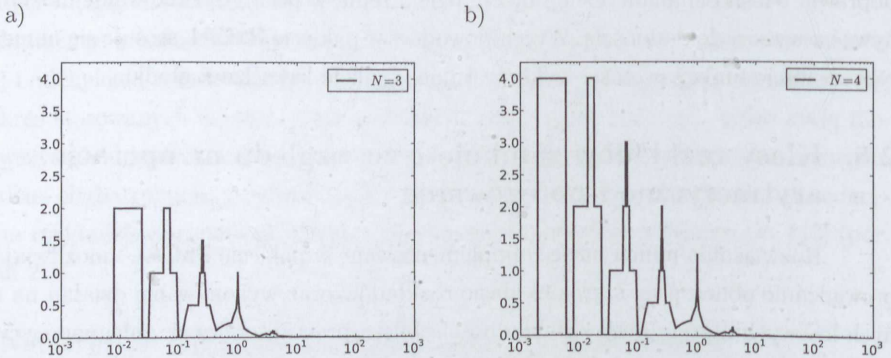
$$f(x) = \sum_{k=0}^{\infty} 2^{-k+1} \chi_{[4^k - \frac{1}{2}, 4^k + \frac{1}{2}]}(x). \quad (2.21)$$

Gęstość ta składa się z nieskończonej liczby prostokątnych impulsów o szerokości równej 1, środkach w punktach $x_k = 4^k$ oraz wysokościach równych $2^{-(k+1)}$ dla $k = 0, 1, \dots$

Ponieważ wysokości impulsów szybko maleją do zera, funkcję f można aproksymować z użyciem funkcji kawałkami wielomianowych w metryce jednostajnej (biorąc na przykład N początkowych składników). Rozważmy iloraz X/Y . Funkcja gęstości ilorazu ma osobliwość zmierzającą do nieskończoności w każdym punkcie postaci $z_k = 4^{-k}$ dla $k = 0, 1, \dots$. Istotnie, niech f_N oznacza N -tą sumę częściową szeregu (2.21), wtedy dla $z_k = 4^{-k}$ mamy:

$$\begin{aligned} \int_{-\infty}^{+\infty} f_N(y z_k) f_N(y) |y| dy &= \int f_N(x) f_N(x \cdot 4^{-k}) |x| dx = \sum_{j=k}^N f_N(4^j) f_N(4^{j-k}) \cdot 4^j \\ &= \sum_{j=k}^N \frac{1}{2^{j+1}} \cdot \frac{1}{2^{j-k+1}} \cdot 4^j = \sum_{j=k}^N 2^{k-2} \\ &= (N - k + 1) 2^{k-2} \rightarrow \infty \text{ dla } N \rightarrow \infty. \end{aligned}$$

Funkcję gęstości ilorazu dla $N = 3, 4$ przedstawiono na rys. 2.3. Widać, że wraz ze wzrostem N impulsy stają się coraz wyższe.



Rys. 2.3. Ilustracja funkcji gęstości z przykładu 2.2 dla: a) $N = 3$, b) $N = 4$; na osi odciętych użyto skali logarytmicznej

Przykład ten pokazuje, że iloraz dwóch funkcji aproksymowalnych może prowadzić do patologicznych funkcji mających nieskończenie wiele punktów osobliwych. Rzecz jasna, funkcje mające nieskończenie wiele osobliwości tego typu nie mogą być aproksymowane w normie $\|\cdot\|_\infty$ z użyciem skończonych reprezentacji wykorzystujących funkcje wielomianowe. Z numerycznego punktu widzenia możliwa jest aproksymacja funkcji mających pojedyncze bieguny. Jednak w takich przypadkach punkty, w których takie osobliwości mogą występować, powinny być wcześniej znane.

W dalszej części $f^{(i)}$ oznacza i -tą pochodną funkcji f , dokładniej $f^{(0)} = f$, $f^{(i)} = (f^{(i-1)})'$, $i = 1, 2, \dots$. Wprowadźmy następującą klasę funkcji $\mathcal{F}_k$:

Definicja 2.1. Mówimy, że $f \in \mathcal{F}_k$, jeżeli istnieją punkty $p_1, \dots, p_n \in \mathbb{R}$ (zwane punktami potencjalnie osobiwymi funkcji f) takie, że dla wszystkich $i = 0, 1, \dots, k$ są spełnione następujące warunki:

A.1. $f^{(i)}$ jest ciągła, z wyjątkiem być może punktów $p_1, \dots, p_n$.

A.2. $|f^{(i)}(x)| = \mathcal{O}(\frac{1}{|x-p_i|^{i+1}})$ dla $x \rightarrow p_i$ dla każdego punktu potencjalnie osobiwego p_i funkcji f .

A.3. $|f^{(i)}(x)| = \mathcal{O}(\frac{1}{|x|^{i+1}})$ dla $x \rightarrow \pm\infty$.

Warunek A.1 mówi, że funkcje z rodziny $\mathcal{F}_k$ są odpowiednio gładkie. Dwa warunki, A.2 oraz A.3, ograniczają tempo zbieżności do nieskończoności oraz do zera w otoczeniu punktów osobiwych. Warunek A.3 jest na przykład spełniony dla wszystkich funkcji całkowicie monotonicznych [51]⁴. Funkcje gęstości niespełniające warunku A.3 muszą zbiegać wolniej do 0 dla $x \rightarrow \infty$ niż funkcja $\frac{1}{x}$ (o ile w ogóle zbiegają), ponieważ całka $\int_a^\infty \frac{1}{x} dx = \infty$. Zatem nie mogą być to funkcje monotoniczne. Podobnie sytuacja wygląda w przypadku warunku A.3. Funkcje rozważane w przykładach: 2.2 oraz 2.3 (podanym w dalszej części) są przykładami funkcji nienależących do rodziny $\mathcal{F}_k$. Zachodzi też oczywiście ciąg inkluzji: $\mathcal{F}_0 \supset \mathcal{F}_1 \supset \dots$.

Celem wprowadzenia rodziny jest zdefiniowanie z jednej strony dość regularnego zbioru funkcji, jednak z drugiej strony wystarczająco obszernego, aby obejmował większość rozkładów stosowanych w praktyce.

Okazuje się, że rodzina funkcji $\mathcal{F}_k$ (przy ustalonym k) jest zamknięta ze względu na probabilistyczne operacje arytmetyczne oraz potęgowanie: Co więcej, działania na funkcjach tego typu umożliwiają ścisłą kontrolę punktów osobiwych w trakcie obliczeń probabilistycznych. Zachodzą następujące twierdzenia [51]:

Twierdzenie 2.1. Niech $f \in \mathcal{F}_k$, $g \in \mathcal{F}_k$ będą funkcjami gęstości prawdopodobieństwa zmiennych niezależnych X i Y , a $p_1, \dots, p_n$ oraz $q_1, \dots, q_m$ będą potencjalnymi punktami osobiwymi, odpowiednio, funkcji f i g . Wtedy:

1. *Suma*: $PDF_{X+Y} \in \mathcal{F}_k$ z potencjalnymi osobiwościami w punktach: $p_i + q_j$, $1 \leq i \leq n$, $1 \leq j \leq m$.

2. *Różnica*: $PDF_{X-Y} \in \mathcal{F}_k$ z potencjalnymi osobiwościami w punktach: $p_i - q_j$, $1 \leq i \leq n$, $1 \leq j \leq m$.

3. *Iloczyn*: $PDF_{X \cdot Y} \in \mathcal{F}_k$ z potencjalnymi osobiwościami w punktach: 0 oraz $p_i q_j$, $1 \leq i \leq n$, $1 \leq j \leq m$.

4. *Iloraz*: $PDF_{X/Y} \in \mathcal{F}_k$ z potencjalnymi osobiwościami w punktach: 0 oraz p_i/q_j , $1 \leq i \leq n$, $1 \leq j \leq m$, $q_j \neq 0$.

⁴ Funkcja f określona na $(0, \infty)$ jest całkowicie monotoniczna, jeżeli $(-1)^n f^{(n)}(x) \geq 0$ dla $x > 0$ [28, podrozdz. XIII.4].

Twierdzenie 2.2. Niech $r \in \mathbb{R}, r \neq 0$ oraz niech X będzie zmienną losową mającą funkcję gęstości $f \in \mathcal{F}_k$ z potencjalnymi punktami osobiowymi $p_1, \dots, p_n, p_i \neq 0$. Wtedy funkcja gęstości zmiennej losowej X^r również należy do $\mathcal{F}_k$ z potencjalnymi punktami osobiowymi $0, p_1^r, \dots, p_n^r$. Jeżeli $r \notin \mathbb{Z}$, zakładamy dodatkowo, że $f(x) = 0$, gdy $x < 0$.

Szczegółowe dowody powyższych twierdzeń są dość długie i w dużej części techniczne, przedstawiono je w pracy [51]. Strategia dowodowa obu twierdzeń jest następująca: Stosunkowo prosto dowodzi się części twierdzenia 2.1 dla sumy (oraz różnicy) zmiennych losowych. Wykorzystywane jest tu twierdzenie o rozkładzie jedności [44, podrozdz. 1.4.], które pozwala rozdzielić w sposób k -gładki funkcje z osobiwościami na część regularną oraz część zawierającą osobiwości. Ponadto własności splotu oraz nierówność Younga pozwalają ograniczyć całkę splotową w terminach norm argumentów. W przypadku iloczynu zmiennych wykorzystuje się standardowe przekształcenie dla zmiennych dodatnich $X \cdot Y = \exp(\log(X) + \log(Y))$ [113]. Niestety przekształcenie to wymaga oszacowania zachowania się pochodnych k -tego rzędu funkcji typu $f(e^x)$ czy $f(\log x)$. W tym celu wykorzystano formułę Faà di Bruno na pochodną k -tego rzędu funkcji złożonej [57]. Zastosowanie tej formuły do funkcji $f(x^\alpha)$ pozwala oszacować funkcję gęstości potęgi zmiennej losowej i dowieść twierdzenia 2.2, które następnie jest wykorzystywane w końcówce dowodu twierdzenia 2.1, w części dotyczącej ilorazu. Wykorzystuje się tu tożsamość $X/Y = X \cdot (Y^{-1})$, która sprowadza dzielenie do iloczynu oraz potęgowania jednej ze zmiennych, a na mocy twierdzenia 2.2 zmienna Y^{-1} spełnia założenia twierdzenia 2.1.

Dodanie w definicji rodziny $\mathcal{F}_k$ pochodnych wyższego rzędu może wydawać się sztuczne oraz znacznie komplikujące część dowodową. Jednak gwarancje wynikające z tych twierdzeń mają duże znaczenie dla zapewnienia jakości aproksymacji. Znany jest fakt, że jakość aproksymacji rośnie wraz rzędem gładkości (liczbą ciągłych pochodnych) funkcji (patrz podrozdz. 4.5).

Ponadto twierdzenia te pokazują, że przy pewnych, dość łagodnych założeniach można efektywnie kontrolować miejsca, w których, w wyniku operacji arytmetycznych na zmiennych losowych, mogą pojawić się osobiwości. W takich punktach można zastosować specjalne metody aproksymacji funkcji z osobiwościami w celu poprawy dokładności. Istnienie rodzin $\mathcal{F}_k$ dowodzi praktycznej możliwości implementacji numerycznej operacji arytmetycznych na stosunkowo obszernym zbiorze niezależnych zmiennych losowych [51]. Rodziny $\mathcal{F}_k$ zawierają praktycznie wszystkie rozkłady znane z zastosowań.

Warto zwrócić uwagę na fakt, że na podstawie definicji 2.1 można wprowadzić klasę funkcji $\mathcal{F}_k^0$ potencjalnie mających osobiwość jedynie w zerze. Jest to w pewnym sensie najbardziej naturalna klasa rozkładów. Klasa ta jest oczywiście zamknięta ze względu na operacje arytmetyczne oraz potęgowanie, ale jeżeli ograniczamy się zgodnie z założeniami twierdzeń do zmiennych ciągłych, to osobiwości poza zerem mogą pojawiać się w wyniku

przesuwania zmiennych losowych (dodawania zmiennych o rozkładach dyskretnych) lub w wyniku wykonywania na zmiennych losowych bardziej złożonych funkcji.

Z obu twierdzeń widać ponownie, że zero jest punktem wyróżnionym operacji mnożenia, dzielenia oraz potęgowania. Twierdzenia te pokazują, że osobliwość w zerze może pojawić się nawet w przypadku, jeżeli argumenty nie mają punktów osobliwych, na co znane są stosowne przykłady (jak rozkład χ^2 z jednym stopniem swobody – kwadrat rozkładu normalnego czy iloczyn i iloraz rozkładu Cauchy’ego [113]). Jednak twierdzenia te mówią, że poza zerem w takim przypadku osobliwości na pewno się nie pojawiają. Warto też zwrócić uwagę, że w wyniku mnożenia lub dzielenia osobliwości mogą, ale nie muszą się pojawić.

Konstruując przykład 2.2 oraz mając na uwadze twierdzenie 2.1, należało znaleźć odpowiednią funkcję spoza rodziny $\mathcal{F}_k$. Istotnie, zauważmy, że wartości funkcji f (z przykładu 2.2) w punktach $x = 4^{-k}$, $k = 1, 2, \dots$ są równe: $f(4^{-k}) = 2^{-(k+1)} = \frac{1}{2}\sqrt{x}$, a więc można jedynie twierdzić, że $f = \mathcal{O}(\sqrt{x})$ dla $x \rightarrow \infty$, czyli f nie należy do $\mathcal{F}_k$. Jasne jest, że funkcję f można zmodyfikować w ten sposób, aby była on ciągła lub k -gładka. Twierdzenie 2.1 mówi, że ograniczając się do funkcji spełniających warunki A.1–A.3, chronimy się przed pojawianiem się patologii jak w opisanym przykładzie oraz użycie tych założeń jest istotne.

Twierdzenie 2.2 mówi, że rodzina $\mathcal{F}_k$ jest zamknięta ze względu na złożenia z funkcjami potęgowymi. Poniższy przykład pokazuje, że złożenia zmiennych losowych z funkcjami innymi niż potęgowe należy również wykonywać ostrożnie.

Przykład 2.3 (logarytm a zbiory $\mathcal{F}_k$). Niech X będzie zmienną losową o rozkładzie danym szeregiem

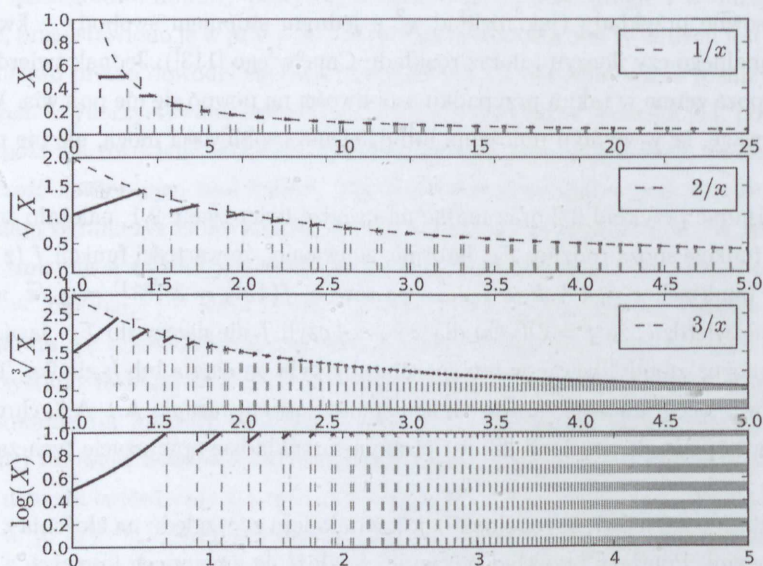
$$f(x) = \sum_{i=1}^{\infty} \frac{1}{i+1} \chi_{[i+1-\frac{1}{i}, i+1]}(x).$$

Rozkład ten składa się z nieskończonej liczby prostokątów o szerokości $\frac{1}{i}$ i wysokości $\frac{1}{i+1}$, $i = 1, 2, \dots$. Ponieważ $f(x) = \frac{1}{i+1}$ dla $x = i+1$ oraz $f(x) \leq \frac{1}{i+1}$ dla $x \in [i, i+1]$, zatem mamy $f(x) = \mathcal{O}(\frac{1}{x})$. Na podstawie (2.5) dla $y \in [\log(j+1 - \frac{1}{j}), \log(j+1)]$ mamy:

$$\begin{aligned} PDF_{\log(X)}(y) &= \sum_{i=1}^{\infty} \frac{1}{i+1} \chi_{[i+1-\frac{1}{i}, i+1]}(\exp y) \exp(y) \\ &= \frac{1}{j+1} \exp(y) \leq \frac{1}{j+1} \exp(\log(j+1)) = 1, \end{aligned}$$

a równość zachodzi dla $y = \log(j+1)$, zatem $PDF_{\log(X)} \notin \mathcal{O}(\frac{1}{x})$, czyli $PDF_{\log(X)} \notin \mathcal{F}$. Natomiast z twierdzenia 2.2 wynika, że $\sqrt{X}$, $\sqrt[3]{X} \in \mathcal{F}$. Początkowe fragmenty funkcji gęstości zmiennych losowych X , $\sqrt{X}$, $\sqrt[3]{X}$, $\log(X)$, wraz z teoretycznymi obwiedniami, przedstawiono na rys. 2.4.

Widać stąd, że logarytmowanie dla pewnych zmiennych losowych może wyprowadzać wynik poza rodzinę $\mathcal{F}$, dalsze operacje na tego typu funkcjach mogą powodować trudności przy aproksymacji wyniku.



Rys. 2.4. Rozkład zmiennej losowej X oraz zmiennych $\sqrt{X}$, $\sqrt[3]{X}$, $\log(X)$

3. Obliczenia na zmiennych losowych zależnych

3.1. Wstęp

W rozdziale tym zostaną omówione wybrane aspekty obliczeń na zmiennych losowych zależnych. Pokróćce zostaną przedstawione wielowymiarowe modele zależności oraz podstawowe miary zależności pomiędzy zmiennymi. Całość rozważań dotyczy zmiennych ciągłych.

Wielowymiarowe modele probabilistyczne można zdefiniować, używając wielu różnych metod. Najprostszy przypadek polega na określeniu łącznego rozkładu wszystkich zmiennych jawną postacią funkcyjną, jak na przykład wielowymiarowy rozkład normalny o zadanych parametrach. Bardziej złożonym i ogólniejszym modelem zależności są modele wykorzystujące tak zwane kopuły [55, 89]. Kolejnym bardziej złożonym modelem są sieci bayesowskie [53, 96, 97], które umożliwiają definiowanie wielowymiarowego rozkładu łącznego z wykorzystaniem niskowymiarowych rozkładów warunkowych oraz formy produktowej.

Zasadniczą operacją numeryczną wykonywaną w trakcie wnioskowania jest całkowanie (lub sumowanie w przypadku dyskretnym). Produktowa postać rozkładu łącznego znacznie pomaga przy liczeniu całek, gdyż pozwalają one w niektórych przypadkach obniżyć wymiarowość problemu, wyłączając czynniki niezależne od zmiennych, względem których odbywa się całkowanie. Postać funkcyjna modelu w niektórych przypadkach (na przykład rozkład normalny) pozwala uprościć niektóre zadania wnioskowania do tego stopnia, że całkowanie nie jest potrzebne. W ogólności jednak, gdy występują zależności, proces wnioskowania wymaga całkowania funkcji wielowymiarowych oraz przechowywania wyników – w ogólnym przypadku rozkładów wielowymiarowych. Dokładne omówienie ogólnego wnioskowania probabilistycznego w modelach graficznych wykracza poza zakres niniejszej monografii. W praktycznej części ograniczono się jedynie do modeli danych jawnie funkcją gęstości oraz modeli definiowanych za pomocą kopuł. Ponadto większy nacisk został położony na wykonywanie obliczeń na zmiennych zależnych oraz związane z tym transformacje zmiennych.

3.2. Wnioskowanie probabilistyczne

3.2.1. Sformułowanie zagadnienia

Samo sformułowanie zadania wnioskowania probabilistycznego jest raczej proste („trywialne”, dokładnie cytując J. Pearl’a [96, s. 20]). W ustalonym modelu probabilistycznym (opisanym wielowymiarowym rozkładem łącznym) ustalamy dwa podzbiory zmiennych: zbiór zmiennych prognozowanych $Y \subset V$ oraz zbiór zmiennych, na których będą określane warunki (zmiennych warunkowych) $X \subset V$, gdzie V jest zbiorem wszystkich zmiennych występujących w modelu. Wyznaczenie rozkładu warunkowego $P(X|Y)$ w przypadku dyskretnym polega na zastosowaniu sumowania:

$$P(Y|X) = \frac{\sum_S P(X, Y, S)}{\sum_{Y, S} P(X, Y, S)}, \quad (3.1)$$

gdzie sumowanie przebiega po zbiorze $S = V \setminus (X \cup Y)$ wszystkich pozostałych zmiennych występujących w modelu. W przypadku ciągłym sumy muszą zostać zastąpione całkowaniem:

$$P(Y|X) = \frac{\int_S P(X, Y, S)}{\int_{Y, S} P(X, Y, S)}. \quad (3.2)$$

W tym miejscu widoczna jest podstawowa trudność obliczeniowa – liczenie całek wielowymiarowych (podobnie zresztą jak sum) jest numerycznie złożone. Warto zwrócić uwagę, że całki te zależą od parametrów – zmiennych ze zbioru X i Y , czyli wynik całkowania jest funkcją tych parametrów (zmiennych). Wnioskowanie probabilistyczne zasadniczo wymaga wykonania dwóch kroków:

- *warunkowania* – jest to podstawienie warunków na zmienne ze zbioru X oraz normalizacja rozkładu, co w ogólności również wymaga całkowania;
- *eliminacji zmiennych* (lub *marginalizacji*) – jest to usunięcie z modelu pozostałych, niepotrzebnych zmiennych ze zbioru S ; wykonuje się to zwykle przez całkowanie lub sumowanie.

W niektórych przypadkach zmienne prognozowane mogą być powiązane deterministycznymi równaniami strukturalnymi (ang. *structural equations*) [96] z pozostałymi zmiennymi. W takich przypadkach przed przystąpieniem do warunkowania lub marginalizacji konieczne jest wykonanie transformacji zmiennych. Ponadto nie każdy rodzaj wnioskowania wymaga warunkowania czy marginalizacji.

3.2.2. Stosowane rozwiązania

W przypadku obliczania sum typu (3.1) widać, że nawet dla n dyskretnych zmiennych binarnych (przyjmujących dwie wartości) przechowywanie rozkładu łącznego w ogólności wymaga 2^n komórek pamięci. Marginalizacja takich rozkładów może wymagać liczby

sumowań porównywalnej do 2^n , co jest wykładniczo duże wraz z liczbą zmiennych. Podobnie ma się sytuacja ze zmiennymi ciągłymi.

Standardowo stosowane rozwiązania ogólne korzystają generalnie z dwóch ważnych koncepcji:

- *faktoryzacji rozkładu* – co pozwala na obniżenie wymiarowości problemu [97],
- *liczenia całek postaci (3.2) z wykorzystaniem metody Monte Carlo* [40, 80, 81, 88].

Faktoryzacja rozkładu jest szeroko wykorzystywana w sieciach bayesowskich i modelowaniu graficznym [54, 96, 97]. Dokładne wnioskowanie w sieciach bayesowskich jest możliwe jedynie w przypadku zmiennych dyskretnych oraz w szczególnych przypadkach, jak zmienne ciągłe opisane wielowymiarowym rozkładem normalnym [11]. Dokładne metody wnioskowania dla dowolnego typu zmiennych nie są obecnie znane, powszechne jest natomiast stosowanie metod przybliżonych opartych na próbkowaniu Gibbsa oraz łańcuchach Markowa [2, 78, 88]. Nie spotyka się obecnie ogólnych systemów wnioskowania, które wykorzystują reprezentacje numeryczne oraz wykonują operacje probabilistyczne w sposób numeryczny. Natomiast można spotkać pewne próby wykorzystujące obliczenia symboliczne.

W przypadku modeli z małą liczbą zmiennych można się posługiwać bezpośrednio rozkładem łącznym w postaci funkcyjnej. Proces wnioskowania w modelach niskowymiarowych może odbywać się przez bezpośrednie całkowanie. Natomiast w niektórych przypadkach również warunkowanie można nieco uprościć. W dalszej części zostaną przedstawione wybrane modele zmiennych zależnych oraz wybrane schematy obliczeniowe na zmiennych zależnych.

3.3. Wielowymiarowe modele zależności

3.3.1. Wielowymiarowy rozkład normalny

Funkcja gęstości wielowymiarowego rozkładu normalnego ma postać:

$$f(\mathbf{x}) = \frac{1}{\sqrt{2\pi^n \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right),$$

gdzie μ – jest wektorem wartości średnich, a Σ – jest macierzą kowariancji [50].

Wnioskowanie probabilistyczne na zmiennych opisanych wielowymiarowym rozkładem normalnym w pewnych przypadkach upraszcza się znacznie. Wynika to z własności rozkładu normalnego. Na przykład rozkłady warunkowe oraz brzegowe są, w takim przypadku, rozkładami normalnymi o znanych postaciach wektora średnich i macierzy kowariancji [11, podrozdz. 2.3], co więcej, parametry te można wyznaczyć, nie odwołując się do całkowania. Podobnie transformacje afiniczne zmiennych opisanych łącznym rozkładem normalnym prowadzą ponownie do zmiennych o rozkładzie normalnym [11, podrozdz. 2.3, 8.1.4].

3.3.2. Kopuły

Kopuły [55, 89] są wygodnym sposobem modelowania zależności pomiędzy zmiennymi losowymi. Znając rozkłady brzegowe zmiennych oraz pewne ilościowe miary zależności pomiędzy zmiennymi, można z użyciem kopuł modelować rozkłady łączne spełniające założone warunki. Rozważania w tym podrozdziale ograniczone są do przypadku dwuwymiarowego, lecz większość definicji i niektóre własności są poprawne dla większej liczby wymiarów.

Definicja 3.1. Dwuwymiarową kopułą nazywamy funkcję C określoną na $[0, 1] \times [0, 1]$ spełniającą następujące warunki:

1. *Warunki na brzegu:* dla każdego $u, v \in [0, 1]$

$$C(u, 0) = C(0, v) = 0,$$

$$C(u, 1) = u \text{ oraz } C(1, v) = v.$$

2. *2-monotoniczność:* dla każdego $u_1, u_2, v_1, v_2 \in [0, 1]$ takiego, że $u_1 \leq u_2$ oraz $v_1 \leq v_2$, zachodzi:

$$C(u_2, v_2) - C(u_2, v_1) - C(u_1, v_2) + C(u_1, v_1) \geq 0. \quad (3.3)$$

Wyrażenie występujące po lewej stronie nierówności (3.3) można przepisać, używając operatora różnicy $\Delta_{v_1}^{v_2} F(u, v) = F(u, v_2) - F(u, v_1)$, o następującej postaci:

$$\begin{aligned} & C(u_2, v_2) - C(u_2, v_1) - C(u_1, v_2) + C(u_1, v_1) \\ &= (C(u_2, v_2) - C(u_2, v_1)) - (C(u_1, v_2) - C(u_1, v_1)) = \Delta_{v_1}^{v_2} C(u_2, v) - \Delta_{v_1}^{v_2} C(u_1, v) \\ &= \Delta_{u_1}^{u_2} \Delta_{v_1}^{v_2} C(u, v) = V_C([u_1, u_2] \times [v_1, v_2]). \end{aligned}$$

Jeżeli $C(u, v) = uv$, to wyrażenie V_C mierzy pole prostokąta zgodnego z osiami układu współrzędnych zbudowanego na wierzchołkach (u_1, v_1) oraz (u_2, v_2) . W ogólnym przypadku funkcję C spełniającą warunek (3.3) nazywamy 2-monotoniczną, natomiast V_C nazywamy C -objętością. Użyteczność kopuł w rachunku prawdopodobieństwa pokazuje następujące twierdzenie [89, podrozdz. 2.3]:

Twierdzenie 3.1 (A. Sklar, 1959). Niech H będzie łączną dystrybuantą zmiennych X i Y , a F i G będą, odpowiednio, dystrybuantami brzegowymi. Wtedy istnieje kopuła C taka, że

$$H(x, y) = C(F(x), G(y)). \quad (3.4)$$

Jeżeli F i G są ciągle, to C jest określona jednoznacznie. Z drugiej strony, jeżeli C jest kopułą, to funkcja H zdefiniowana przez (3.4) określa dystrybuantę pewnego rozkładu dwuwymiarowego.

Zasadniczą zaletą w stosowaniu kopuł jest to, że pozwalają rozdzielić jawnie rozkłady zmiennych (rozkłady brzegowe) oraz sam model zależności pomiędzy zmiennymi określony przez kopułę. Istnienie takiego rozdzielania nie jest od razu widoczne, gdy posługujemy się jedynie rozkładem łącznym¹. W takim przypadku rozkłady brzegowe poszczególnych zmiennych trzeba wyznaczać przez marginalizację rozkładu (całkując względem pozostałych zmiennych).

W szczególnym przypadku, gdy $F(u) = u$ i $G(v) = v$ dla $u, v \in [0, 1]$, kopuła opisuje dystrybuantę pewnego rozkładu łącznego o jednostajnych rozkładach brzegowych.

Niech $X_i \sim f_i, F_i$ dla $i = 1, \dots, n$ będą zmiennymi losowymi, a f i F będą, odpowiednio, ich łączną funkcją gęstości i łączną dystrybuantą. Kopuła C definiuje dystrybuantę zmiennej losowej. Jeżeli dystrybuanta ta jest odpowiednio regularna, to możemy wyznaczyć funkcje gęstości przez różniczkowanie względem kolejnych zmiennych, otrzymując kopułową funkcję gęstości:

$$c(u_1, \dots, u_n) = \frac{\partial^n}{\partial u_1 \dots \partial u_n} C(u_1, \dots, u_n)$$

oraz podobnie funkcję gęstości:

$$f(u_1, \dots, u_n) = \frac{\partial^n}{\partial u_1 \dots \partial u_n} F(u_1, \dots, u_n) = c(u_1, \dots, u_n) f_1(u_1) \cdot \dots \cdot f_n(u_n).$$

Podobnie wyznacza się rozkłady warunkowe; na przykład w przypadku dwuwymiarowym mamy:

$$c(u_1|u_2) = \frac{\partial C(u_1, u_2)}{\partial u_2},$$

$$f(u_1|u_2) = c(u_1|u_2) f_2(u_2).$$

Rozkłady warunkowe wykorzystuje się zarówno do samego wnioskowania (przy wyznaczaniu rozkładów warunkowych), jak i do generowania próbek losowych z rozkładu łącznego opisanego za pomocą kopuł [89, podrozdz. 2.9].

Dla danej kopuły C określa się kopułę dualną (ang. *dual copula*):

$$\bar{C}(u, v) = u + v - C(u, v)$$

¹ Nazwa „kopuła” pochodzi od słów *copula*, *copule* oznaczających łącznik lub złączkę i nie powinno być mylone z wyrazem „kopuła” oznaczającym wypukłą część sklepienia (ang. *dome*). Twierdzenie Sklara pokazuje, że kopuły pełnią funkcję łącznika pomiędzy rozkładami brzegowymi.

oraz kopułę przeżycia², zdefiniowaną jako:

$$\hat{C}(u, v) = u + v - 1 + C(1 - u, 1 - v).$$

Ponadto definiuje się również ko-kopułę (ang. *co-copula*):

$$C^*(u, v) = 1 - C(1 - u, 1 - v).$$

Ograniczenia Fréchet–Hoeffdinga, kopuły: Π , M , W ; T -normy

Ograniczenia Fréchet–Hoeffdinga (ang. *Fréchet–Hoeffding bounds*) mówią [89, podrozdz. 2.2], że wszystkie kopuły są ograniczone z dołu przez W -kopułę i z góry przez M -kopułę w następujący sposób:

$$W(u_1, \dots, u_n) \leq C(u_1, \dots, u_n) \leq M(u_1, \dots, u_n),$$

gdzie:

$$W(u_1, \dots, u_n) = \max \left(-n + 1 + \sum_{i=1}^n u_i, 0 \right),$$

$$M(u_1, \dots, u_n) = \min(u_1, \dots, u_n).$$

Oprócz kopuł W i M wyróżnione miejsce zajmuje kopuła produktowa (Π -kopuła):

$$\Pi(u_1, \dots, u_n) = u_1 \cdot \dots \cdot u_n.$$

Kopuły te przedstawiono na rys. 3.1.

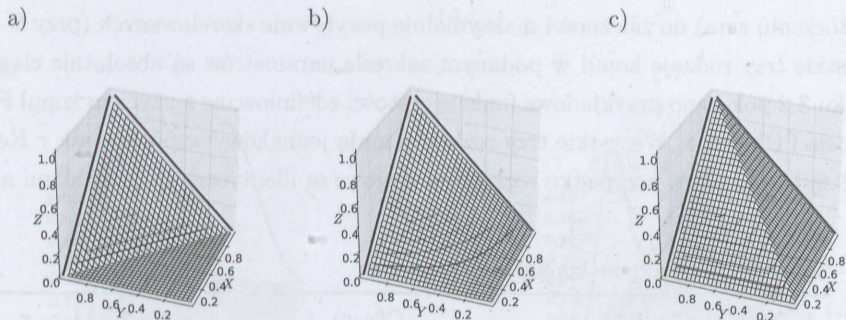
Pokrewnym pojęciem związanym z kopułami są T -normy (lub normy trójkątne) [25].

Definicja 3.2. T -normą nazywamy funkcję $T: [0, 1] \times [0, 1] \rightarrow [0, 1]$ spełniającą następujące warunki:

1. *Przemienność*: $T(x, y) = T(y, x)$.
2. *Łączność*: dla dowolnych x, y, z zachodzi $T(x, T(y, z)) = T(T(y, x), z)$.
3. *Monotoniczność*: jeżeli $x \leq y$, to $T(x, y) < T(y, z)$.
4. *Warunki na brzegu*: $T(x, 1) = T(1, x) = x$, $T(x, 0) = T(0, x) = 0$.

Z porównania obu definicji widać, że od T -norm nie wymaga się 2-monotoniczności, ale jedynie monotoniczności ze względu na poszczególne argumenty, natomiast od kopuł nie wymaga się symetrii i łączności. Wiele jednak kopuł, jak na przykład: M , Π czy W ,

² Analogicznie do funkcji przeżycia (ang. *survival function*), którą dla zmiennej losowej $X \sim f$, F definiuje się jako $\Pr(X > x) = \int_x^\infty f(x) dx = 1 - F(x)$. W przypadku gdy zmiennej x nie można interpretować jako czasu, używa się też pojęcia dystrybucyjnego dopełniającego (ang. *complementary cumulative distribution function*).



Rys. 3.1. Szczególne rodzaje kopuł: a) W, b) II, c) M

jest jednocześnie normami trójkątnymi. Przeglądowe porównanie i dokładniejsze omówienie wzajemnych relacji pomiędzy kopułami a T -normami można znaleźć w pracy [86]. T -normy są wykorzystywane jako uogólnienie operatora koniunkcji w logice rozmytej [98].

3.3.3. Kopuły archimedesowe

Kopuły archimedesowe są to kopuły dające przedstawić się w postaci:

$$C(u, v) = \begin{cases} \varphi^{-1}(\varphi(u) + \varphi(v)) & \text{dla } \varphi(u) + \varphi(v) \leq \varphi(0), \\ 0 & \text{w pozostałych przypadkach} \end{cases}$$

dla pewnej funkcji φ , która nosi nazwę generatora. Dokładne warunki, jakie musi spełniać generator, aby definiował kopułę, można znaleźć w pracach [55, 89]. Na przykład φ jest generatorem, jeżeli spełnia następujące warunki:

1. $\varphi \in C^2[0, 1]$, $\varphi(1) = 0$.
2. $\varphi'(u) < 0$, dla $u \in [0, 1]$.
3. $\varphi''(u) > 0$, dla $u \in [0, 1]$.

Znane są różne rodziny kopuł różniących się generatorem zależnym od pojedynczego parametru [89, rozdz. 4].

Przykładowe kopuły mające większe znaczenie w zastosowaniach zebrano w tab. 3.1. Więcej przykładów można znaleźć w pracach [55, 89]. Parametr θ jest powiązany z siłą zależności pomiędzy zmiennymi. Parametr ten przekłada się bezpośrednio na współczynnik τ Kendalla (tab. 3.2, podrozdz. 3.4).

Kopuły Clayтона i Gumbela umożliwiają modelowanie pozytywnych korelacji w pełnym zakresie zmienności – od braku korelacji (dla wartości θ bliskich, odpowiednio, 0 i 1) do pełnej zależności funkcyjnej (dla dużych wartości θ) w granicy opisanej kopułą M . Kopuła Franka umożliwia modelowanie zależności w pełnym zakresie – od maksymalnie negatywnych (przy $\theta \rightarrow -\infty$) przez zależności odpowiadające zmiennym niezależnym (dla

θ w otoczeniu zera) do zależności maksymalnie pozytywnie skorelowanych (przy $\theta \rightarrow \infty$). Wszystkie trzy rodzaje kopuł w podanym zakresie parametrów są absolutnie ciągłe. Na rysunku 3.2 pokazano przykładowe funkcje gęstości zdefiniowane z użyciem kopuł Franka, Gumbela i Claytona. Wszystkie trzy rozkłady mają jednakowy współczynnik τ Kendalla ($\tau = 0.66$), w każdym przypadku rozkłady brzegowe są identycznymi rozkładami normalnymi.

Tab. 3.1. Przykładowe kopuły archimedesowe

Kopuła *	$\varphi(t; \theta)$	$C(u, v)$	θ	τ
Π	$-\log t$	uv	—	0
Claytona	$\frac{1}{\theta}(t^{-\theta} - 1)$	$(u^{-\theta} + v^{-\theta} - 1)^{-1/\theta}$	$(0, \infty)$	$1 - \frac{2}{2+\theta}$
Gumbela	$(-\log(t))^{\theta}$	$\exp\left(-\left((-\log u)^{\theta} + (-\log v)^{\theta}\right)^{\frac{1}{\theta}}\right)$	$(0, \infty)$	$1 - \frac{1}{\theta}$
Franka	$-\log \frac{e^{-\theta t} - 1}{e^{-\theta} - 1}$	$-\log\left(1 + \frac{(\exp(\theta u) - 1)(\exp(\theta v) - 1)}{\exp(\theta) - 1}\right)$	$\mathbb{R} \setminus \{0\}$	— **

* Na podstawie [89, rozdz. 4].

** Zwarta postać wymaga funkcji specjalnych [46].

Kopuły zagnieżdżone

Jednoparametrowe kopuły archimedesowe opisują dość proste zależności, które w zasadzie dobrze nadają się do modelowania zależności dwuwymiarowych. Do opisu zależności w ogólnych modelach wielowymiarowych potrzebna jest zwykle większa liczba parametrów³. Na podstawie jednoparametrowych kopuł archimedesowych można tworzyć bardziej złożone modele, zagnieżdżając w sobie kolejne modele dwuwymiarowe [41, 89].

Implementacja kopuł

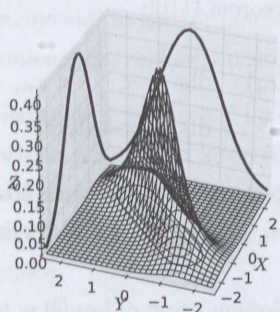
Z technicznego punktu widzenia implementacja kopuł wymaga reprezentacji zarówno samej kopuły, jak i kopułowej funkcji gęstości oraz rozkładów warunkowych. Ponadto definiując rozkłady brzegowe, należy określić dystrybuantę, jak i funkcję gęstości (funkcje te są dostępne w pakiecie **PaCAL**). Jeżeli dodatkowo wymagane jest generowanie liczb losowych z rozkładu zadanego kopułą, to powinna zostać określona odwrotna funkcja gęstości [41, 89].

Implementację kopuł szczególnej postaci można częściowo uprościć. Dla kopuł archimedesowych wystarcza określenie generatora. Wszystkie pozostałe potrzebne funkcje można wyznaczyć na drodze przekształceń symbolicznych (takie rozwiązanie jest obecnie stosowane w pakiecie **PaCAL**) lub numerycznych.

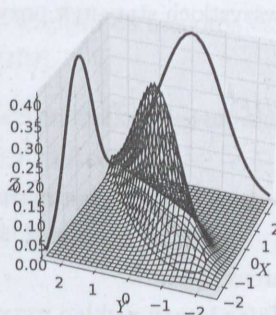
Z numerycznego punktu widzenia istnieje również zasadnicza różnica pomiędzy kopułami osobliwymi, takimi jak W i M , oraz kopułami absolutnie ciągłymi (mającymi funkcję gęstości, jak Π czy kopuła Franka). Jednolite metody numeryczne obliczeń na

³ Na przykład w przypadku wielowymiarowego rozkładu normalnego zależności są określone za pomocą macierzy korelacji.

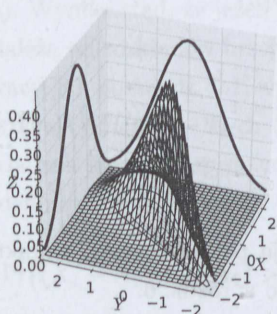
a)



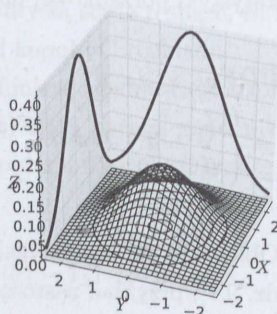
b)



c)



d)



Rys. 3.2. Dwuwymiarowe funkcje gęstości zdefiniowane za pomocą kopuł z normalnymi rozkładami brzegowymi: a) Franka, b) Gumbela, c) Claytona, d) kopuły produktowej Π ; wszystkie rozkłady (z wyjątkiem produktowego) mają jednakową wartość współczynnika $\tau = 0.66$ oraz identyczne rozkłady brzegowe

zmiennych losowych zależnych oparte na całkowaniu dotyczą w zasadzie przypadku absolutnie ciągłego, natomiast rozkłady osobliwe (singularne) względem miary Lebesgue'a w $\mathbb{R}^n$ muszą być traktowane w odmienny sposób.

3.3.4. Próba uporządkowana, łączny rozkład dwóch statystyk pozycyjnych

Kopuły są uniwersalnym narzędziem modelowania zależności pomiędzy zmiennymi, jednak w pewnych przypadkach w naturalny sposób pojawiają się pewne szczególne rodzaje zależności. Załóżmy, że mamy do czynienia z n -elementową próbą losową $X_1, \dots, X_n$, pobraną w sposób niezależny z rozkładu f . Stosunkowo prosto jest wyznaczyć jednowymiarowe rozkłady statystyk pozycyjnych $X_{(i)}$, $i = 1, \dots, n$ (podrozdz. 2.5). Statystyki pozycyjne są jednak wzajemnie zależne. Zależność ta jest spowodowana faktem, że $X_{(i)} < X_{(j)}$,

o ile $i < j$, i jest tym silniejsza, im pozycje są sobie bliższe. Chcąc wykonywać obliczenia na statystykach pozycyjnych, należy znać rozkład łączny [113, podrozdz. 9.7]. Łączny rozkład wszystkich statystyk pozycyjnych jest dany wzorem [110]:

$$PDF_{X_{(1)}, \dots, X_{(n)}}(x_1, \dots, x_n) = \begin{cases} \frac{1}{n!} f(x_1) \cdot \dots \cdot f(x_n) & \text{dla } x_1 < \dots < x_n, \\ 0 & \text{dla pozostałych.} \end{cases}$$

Bierze się to z faktu, że łączny rozkład produktowy jest ograniczony warunkiem $X_{(1)} < X_{(2)} < \dots < X_{(n)}$, natomiast wszystkich możliwych uporządkowań elementów próby n -elementowej jest $n!$.

Mając łączny rozkład wszystkich zmiennych, możemy (przynajmniej w teorii) znaleźć rozkład dowolnego podzioru zmiennych przez całkowanie. W przypadku statystyk pozycyjnych rozkłady łączne można wyznaczyć jawnie. Na przykład łączny rozkład i -tej i j -tej statystyki porządkowej dany jest wzorem [110]):

$$PDF_{X_{(j)}, X_{(k)}}(x, y) = \begin{cases} \binom{n}{i-1, j-i-1, n-j} F(x)^{i-1} (F(y) - F(x))^{j-i-1} (1 - F(y))^{n-j} f(x) f(y), & x \leq y, \\ 0 & x > y. \end{cases}$$

Obliczenia na statystykach pozycyjnych mają pewne znaczenie praktyczne [113, podrozdz. 9.7], przykład zastosowania przedstawiono w rozdz. 6.

3.4. Dwuwymiarowe miary zależności pomiędzy zmiennymi

Rozkład łączny niesie w sobie całą informację na temat zależności pomiędzy zmiennymi. Niemniej w praktyce do oceny siły zależności są wykorzystywane zwykle pewne miary liczbowe, jak współczynnik korelacji. Współczynnik τ Kendalla jest szczególnym przypadkiem ogólnej miary zgodności, konkordancji (ang. *measure of concordance*) wprowadzonej w pracy [107].

Definicja 3.3. Niech X, Y będą zmiennymi losowymi opisanymi kopułą C . Miarę $\kappa_{X,Y}$ nazywamy *miarą zgodności* lub *konkordancją*, jeżeli są spełnione następujące warunki:

1. *Dziedzina*: κ jest określona dla dowolnej pary zmiennych losowych ciągłych.
2. *Zakres*: $-1 \leq \kappa_{X,Y} \leq 1$, $\kappa_{X,X} = 1$.
3. *Symetria*: $\kappa_{X,Y} = \kappa_{Y,X}$.
4. *Niezależność*: jeżeli X i Y są niezależne ($C = \Pi$), to $\kappa_{X,Y} = 0$.
5. *Zamiana znaku*: $\kappa_{X,-Y} = -\kappa(X, Y)$.
6. *Zgodność porządków*: jeżeli $C_{X,Y}^1 \leq C_{V,W}^2$, to $\kappa_{X,Y} \leq \kappa_{V,W}$.

7. *Ciągłość*: niech $(X_n, Y_n) \sim C^n$, $n \in \mathbb{N}$, $(X, Y) \sim C$. Jeżeli ciąg C^n jest punktowo zbieżny do C , to ciąg κ_{X_n, Y_n} jest zbieżny do $\kappa_{X, Y}$.

Okazuje się, że konkordancje są niezmiennicze ze względu na monotoniczne transformacje zmiennych, zachodzi następujące twierdzenie [107]:

Twierdzenie 3.2. Niech f i g będą funkcjami ściśle monotonicznymi określonymi na zbiorze wartości zmiennych losowych, odpowiednio, X i Y . Wtedy:

1. Jeżeli f i g są jednocześnie rosnące lub malejące, to $\kappa_{f(X), g(Y)} = \kappa_{X, Y}$.
2. Jeżeli $Y = f(X)$, to $\kappa_{X, Y} = 1$, gdy f jest funkcją rosnącą, lub $\kappa_{X, Y} = -1$, gdy f jest malejąca.

Należy zwrócić uwagę, że rozkłady definiowane za pomocą kopuł składają się z dwóch części kopuły oraz transformacji monotonicznej zmiennych zdefiniowanej za pomocą dystrybuant rozkładów brzegowych (dystrybuanta zmiennej losowej jest funkcją niemalejącą). Wynika stąd, że jeżeli brzegowe dystrybuanty są ściśle rosnące, to konkordancje nie zależą od rozkładów brzegowych, a jedynie od kopuły C rozkładu.

W pracy [89, podrozdz. 5.1] wprowadzono ogólną funkcję zgodności (konkordancji) (ang. *concordance function*). Niech (X_1, Y_1) oraz (X_2, Y_2) będą parami zmiennych losowych o rozkładach łącznych opisanych, odpowiednio, kopułami C_1 i C_2 . Określimy miarę zgodności zdefiniowaną następująco:

$$Q = \Pr((X_1 - X_2)(Y_1 - Y_2) > 0) - \Pr((X_1 - X_2)(Y_1 - Y_2) < 0).$$

Można pokazać [89, twierdzenie 5.1.1], że miara Q jest zależna jedynie od kopuł C_1 i C_2 oraz jest dana wzorem:

$$Q(C_1, C_2) = 4 \int_0^1 \int_0^1 C_2(u, v) dC_1(u, v) - 1.$$

Ustalając w funkcji Q jeden z argumentów, określamy pewnego rodzaju uporządkowanie w zbiorze wszystkich kopuł. Interesujące jest porównanie zgodności ustalonej kopuły względem charakterystycznych skrajnych kopuł, jak M , W czy Π . W tabeli 3.2 zebrano podstawowe miary konkordancji – współczynniki: τ Kendalla, ρ Spearmana, γ Giniego oraz β Blomqvista (na podstawie [26, 89, 108]).

W pewnych przypadkach miary te można wyznaczyć analitycznie (tab. 3.1), a w każdej sytuacji przez bezpośrednie całkowanie. Obliczanie współczynnika β nie wymaga oczywiście całkowania. W przypadku rozkładów, które nie są absolutnie ciągłe, określenie miar konkordancji nie jest bezpośrednie, niektóre szczegóły można znaleźć w pracy [90].

Współczynnik korelacji liniowej Pearsona nie jest w sensie definicji 3.3 miarą konkordancji. Nietrudno zauważyć, że dla ustalonej kopuły zależy on w mniejszym lub większym stopniu od rozkładów brzegowych. Z tego punktu widzenia jest dość szczególną miarą

zależności charakterystyczną dla zmiennych o rozkładzie normalnym, co przesądza o jego popularności. Współczynnik ten w pełni opisuje zależność pomiędzy zmiennymi jedynie wtedy, gdy wiemy, że rozkład łączny jest wielowymiarowym rozkładem normalnym.

Tab. 3.2. Dwuwymiarowe miary zgodności zmiennych losowych ciągłych (na podstawie [26, 89, 108])

Miara	Wartość w terminach Q	Wartość liczbowa
τ Kendalla	$Q(C, C)$	$4 \int_0^1 \int_0^1 C(u, v) dC(u, v) - 1$
ρ Spearmana	$3Q(C, \Pi)$	$12 \int_0^1 \int_0^1 C(u, v) du dv - 1$
γ Giniego	$Q(C, M) + Q(C, W)$	$2 \int_0^1 \int_0^1 u + v - 1 - u - v dC(u, v) =$ $4 \left(\int_0^1 C(u, u) du + \int_0^1 C(u, 1 - u) du \right) - 2$
β Blomqvista	—	$4C(\frac{1}{2}, \frac{1}{2}) - 1$

3.5. Obliczenia na zmiennych zależnych

3.5.1. Twierdzenie o zamianie zmiennych

W pewnych przypadkach mamy do czynienia ze zmiennymi, o których wiadomo, że są zależne. Na przykład wzrost i waga ustalonej osoby są zwykle w korelacji. Lekarz, aby określić nadwagę danej osoby, posługuje się indeksem masy ciała (ang. *BMI*, *body mass index*) zdefiniowanym jako iloraz wagi w kilogramach dzielony przez kwadrat wzrostu w metrach. Jeżeli interesuje nas rozkład wskaźnika BMI, to musimy wykonać dzielenie dwóch zmiennych, które są zależne.

Podstawowym narzędziem teoretycznym, które umożliwia wykonywanie tego typu obliczeń, jest twierdzenie o zamianie zmiennych pod znakiem całki (lub inaczej twierdzenie o całkowaniu przez podstawianie) [43, 50].

Twierdzenie 3.3. Niech f będzie funkcją n zmiennych określoną na przedziale $G \subset \mathbb{R}^n$, a odwzorowanie $\Phi = [\varphi_1, \dots, \varphi_n]: G \rightarrow \mathbb{R}^n$ będzie różniczkowalne w sposób ciągły na zbiorze G oraz takie, że $\left| \frac{\partial(\varphi_1, \dots, \varphi_n)}{\partial(x_1, \dots, x_n)} \right| \neq 0$ dla $x \in G$. Wtedy

$$\int_{\Phi(G)} f(y_1, \dots, y_n) dy_1 \dots dy_n = \int_G f(\Phi(x_1, \dots, x_n)) \cdot \left| \frac{\partial(\varphi_1, \dots, \varphi_n)}{\partial(x_1, \dots, x_n)} \right| dx_1 \dots dx_n.$$

Twierdzenie to umożliwia wyznaczenie rozkładu zmiennych poddanych transformacji. Jeżeli znany jest rozkład łączny zmiennych, to działania na tych zmiennych losowych

możemy wykonywać bezpośrednio, używając wzorów:

$$PDF_{X+Y}(z) = \int_{-\infty}^{+\infty} p(x, z-x) dx, \quad (3.5)$$

$$PDF_{XY}(z) = \int_{-\infty}^{+\infty} p(x, z/x) \frac{1}{|x|} dx, \quad (3.6)$$

$$PDF_{X/Y}(z) = \int_{-\infty}^{+\infty} p(yz, y) |x| dy = \int_{-\infty}^{+\infty} p(x, x/z) \frac{|x|}{z^2} dy. \quad (3.7)$$

Powyższe wzory dotyczą sytuacji jedynie dwóch zmiennych zależnych. W przypadku większej liczby zmiennych pojawi się problem obliczeniowy. Gdy wykonujemy działanie na dwóch argumentach, należy mieć rozkład łączny określony na tych argumentach. W przypadku niezależności taki rozkład dostajemy od razu, mnożąc dwa rozkłady brzegowe. Natomiast w sytuacji ogólnej rozkład brzegowy dostajemy przez całkowanie względem pozostałych zmiennych. Wykorzystanie *sieci bayesowskich* do reprezentacji rozkładu łącznego pozwala niekiedy trochę uprościć obliczenia, jednak w ogólnym przypadku wyznaczania rozkładów brzegowych nie da się uniknąć.

Warto zwrócić uwagę, że nawet obliczenia na zmiennych niezależnych mogą prowadzić do zmiennych zależnych, głównie w przypadkach, gdy ta sama zmienna losowa pojawia się wielokrotnie w obliczeniach. Aby to zilustrować, rozważmy następujący przykład:

Przykład 3.1. Niech X, Y, Z będą niezależnymi zmiennymi losowymi o rozkładzie jednostajnym: $X, Z \sim U[\frac{1}{2}, \frac{3}{2}]$, $Y \sim U[\frac{3}{2}, \frac{5}{2}]$. Rozważmy następujące wyrażenia:

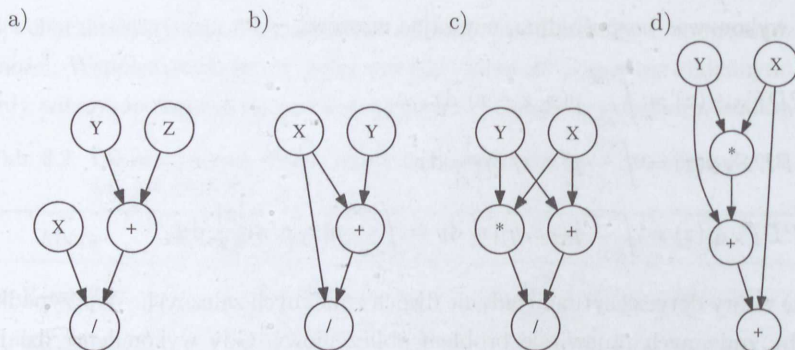
$$\frac{X}{Y+Z}, \quad (3.8)$$

$$\frac{X}{X+Y}, \quad (3.9)$$

$$\frac{XY}{X+Y}, \quad (3.10)$$

$$X + XY + Y. \quad (3.11)$$

W pierwszym przypadku (3.8) zmienne X oraz $Y+Z$ są niezależne i wykonanie dzielenia nie nastręcza trudności, natomiast w drugim przypadku (3.9) zmienne X oraz $X+Y$ są ewidentnie zależne i dzielenie nie może być wykonane z użyciem arytmetyki zmiennych niezależnych, potrzebny jest łączny rozkład zmiennych X i $X+Y$ lub bezpośrednie zastosowanie transformacji zmiennych do wyrażenia $\frac{x}{x+y}$. Podobnie ma się sytuacja z wyrażeniami (3.10) oraz (3.11). Na rysunku 3.3 przedstawiono sieci reprezentujące działania. Jak widać, tylko wyrażeniu (3.8) odpowiada struktura drzewa. Warto zwrócić uwagę, że zależności (3.9), (3.10) i (3.11) można przepisać w takiej postaci, aby każda ze zmiennych



Rys. 3.3. Sieci reprezentujące działania z przykładu 3.1: a) $\frac{X}{Y+Z}$, b) $\frac{X}{X+Y}$, c) $\frac{XY}{X+Y}$, d) $X + XY + Y$

występowała dokładnie jeden raz:

$$\frac{1}{1 + \frac{Y}{X}}, \quad (3.12)$$

$$\frac{1}{\frac{1}{X} + \frac{1}{Y}}, \quad (3.13)$$

$$(X + 1)(Y + 1) - 1. \quad (3.14)$$

Dla tak przekształconych wyrażeń, jeżeli X i Y są niezależne, arytmetykę zmiennych losowych niezależnych można już stosować bezpośrednio.

Automatyzacja tego typu przekształceń jak w przykładzie 3.1 na drodze obliczeń symbolicznych jest w ogólności dość trudna, w niektórych przypadkach konieczne jest też rozwiązywanie równań. Podobny problem występuje w przypadku arytmetyki interwałowej, a pewne metody postępowania z wyrażeniami zawierającymi wielokrotnie tę samą zmienną można znaleźć w pracy [85].

3.5.2. Wyrażenia algebraiczne zawierające dwie zmienne losowe

Niech X i Y będą niezależnymi zmiennymi losowymi o rozkładzie opisanym funkcją gęstości p , a $w(X, Y)$ będzie wyrażeniem algebraicznym wiążącym obie zmienne. Chcąc wyznaczyć rozkład zmiennej losowej $Z = w(X, Y)$, postępujemy analogicznie jak przy wyznaczaniu wzorów (3.5)–(3.7):

1. Definiujemy wartość funkcji gęstości zmiennej Z w punkcie z jako całkę funkcji gęstości wzdłuż konturu $w(x, y) = z$:

$$PDF_Z(z) = \int_{w(x,y)=z} p(x, y) \, dx \, dy.$$

2. Dalej należy rozwikłać funkcję $z = w(x, y)$ względem jednej ze zmiennych, na przykład zmiennej y^4 :

$$y = y_w(x, z).$$

3. W następnym kroku dokonujemy zamiany zmiennych w całce:

$$x = u(x, z) = x,$$

$$y = v(x, z) = y_w(x, z).$$

4. Jakobian przekształcenia wyraża się wzorem:

$$J(x, z) = \begin{vmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial z} \end{vmatrix} = 1 \cdot \frac{\partial y_w(x, z)}{\partial z}.$$

5. Eliminacja zmiennej x prowadzi do całki:

$$PDF_Z(z) = \int_{-\infty}^{+\infty} p(x, y_w(x, z)) J(x, z) dx. \quad (3.15)$$

Bez odwoływania się do obliczeń symbolicznych proces ten jest trudno zautomatyzować, ponieważ wymagałoby to numerycznego odwracania funkcji zależnej od parametru oraz różniczkowania. Zastosowanie obliczeń symbolicznych ograniczone jest natomiast możliwościami bibliotek obliczeń symbolicznych. W bibliotece **PaCAL** procedura ta została zautomatyzowana z wykorzystaniem obliczeń symbolicznych i biblioteki **sympy** [117].

W przypadku większej liczby zmiennych losowych można próbować przekształcić wyrażenie tak, aby można je sprowadzić do przypadku z dwiema zmiennymi lub bezpośrednio do działań na zmiennych niezależnych. Jednak w ogólnej sytuacji, gdy występują więcej niż dwie zmienne opisane rozkładem łącznym, krok eliminacji wymaga liczenia całek wielowymiarowych, co komplikuje się numerycznie wraz z podnoszeniem wymiaru.

Prezentowany schemat obliczeniowy zostanie zilustrowany dwoma przykładami, z których pierwszy jest kontynuacją przykładu 3.1.

Przykład 3.2. Rozważmy zmienną losową Z zdefiniowaną następująco:

$$Z = \frac{X}{X + Y},$$

⁴ Analogicznie można rozwikłać ten kontur ze względu na x lub dokonać dowolnego innego przedstawienia: $x = u(z, t)$, $y = v(z, t)$. Wybór jest zależny oczywiście od postaci wyrażenia $w(x, y)$. Jeżeli wyrażenie w nie jest ustalone przed przystąpieniem do obliczeń, to widzieć od razu, że automatyczny dobór odpowiedniego przekształcenia jest w ogólności trudnym problemem obliczeń symbolicznych.

gdzie zmienne X i Y mają rozkłady jak w przykładzie 3.1. Interesujące nas wyrażenie ma postać:

$$w(x, y) = \frac{x}{x + y}.$$

Rozviklując wyrażenie $w(x, y) = z$ względem y , dostajemy:

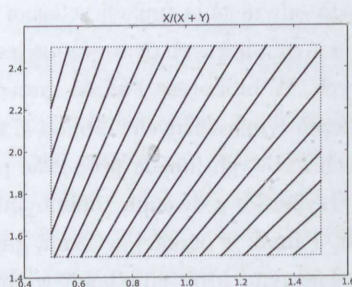
$$y_w(x, y) = \frac{x - xz}{z}.$$

Różniczkując względem z , ostatecznie uzyskujemy jacobian przekształcenia:

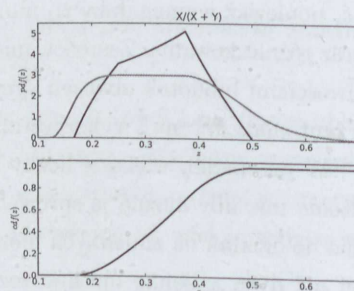
$$J(x, z) = -\frac{x - xz}{z^2} - \frac{x}{z}.$$

Wstawiając do równania (3.15), dostajemy całkę prowadzącą do funkcji gęstości zmiennej losowej Z . Kontury całkowania rozkładu łącznego zmiennych X i Y przedstawiono na rys. 3.4a. Ostatecznie funkcję gęstości oraz dystrybuentę pokazano na rys. 3.4b. Analogiczne wyrugowanie zmiennej x prowadzi do podstawienia: $x_w = \frac{yz}{1-z}$; jacobian przyjmuje wtedy postać $J(y, z) = \frac{y}{1-z} + \frac{yz}{(1-z)^2}$.

a)



b)



Rys. 3.4. Wyniki obliczeń wyrażenia $\frac{X}{X+Y}$ z przykładu 3.2: a) kontury całkowania rozkładu łącznego, b) funkcja gęstości i dystrybuenta zmiennej $U = \frac{X}{X+Y}$; przedstawiono również porównanie obliczeń dla zmiennej $V = \frac{X}{Y+Z}$ z przykładu 3.1

Kolejny, bardziej praktyczny przykład dotyczy wyznaczania rezystancji zastępczej układu rezystorów.

Przykład 3.3 (równoległe połączenie rezystorów). Z fizyki wiadomo, że rezystancja zastępcza równoległego połączenia dwóch rezystorów jest dana wzorem:

$$R = \frac{R_1 R_2}{R_1 + R_2}.$$

Przypuśćmy, że $R_1 = 1 \pm 0.5 \Omega$, $R_2 = 2 \pm 0.5 \Omega$. Chcemy wyznaczyć prawdopodobieństwo tego, że rezystancja zastępcza ma wartość R mniejszą niż 0.5Ω lub większą niż 0.75Ω . W tym celu znajdujemy rozkład rezystancji zastępczej R . Rozviklując wyrażenie $\frac{xy}{x+y} = z$, dostajemy:

$$y_w(x, z) = \frac{xz}{x - z},$$

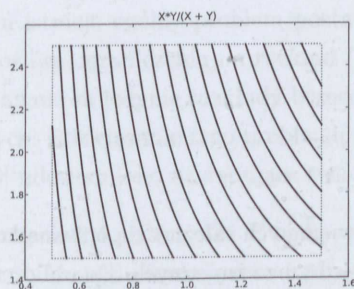
$$J(x, z) = \frac{\partial y_w}{\partial z} = \frac{x}{x - z} + \frac{xz}{(x - z)^2}.$$

Ponownie wstawiając do wzoru (3.15), dostajemy szukany rozkład. Oto podstawowe parametry tego rozkładu:

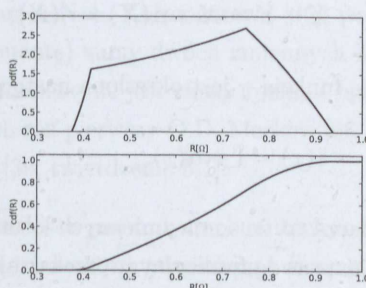
```
mean    = 0.651056167265
median  = 0.659325350484
range   = (0.375, 0.9375)
std     = 0.134701459257
var     = 0.0181444831261
```

Ostatecznie dostajemy rozkład rezystancji zastępczej przedstawiony na rys. 3.5. Mając określoną funkcję gęstości, można obliczyć, że $\Pr(R < 0.5 \Omega) = 0.1732868$ oraz $\Pr(R > 0.75 \Omega) = 0.27339495$.

a)



b)



Rys. 3.5. Rozkład rezystancji zastępczej dwóch rezystorów (przykład 3.3): a) kontury całkowania, b) funkcja gęstości wraz dystrybucją

3.5.3. Skrajne zależności pomiędzy zmiennymi, kopuły M i W

Kopuła Franka umożliwia modelowanie dwuwymiarowych rozkładów zmiennych zależnych w pełnym zakresie zmienności współczynnika τ Kendalla, pomijając skrajne przypadki odpowiadające wartościom $+1$ i -1 . Ponieważ kopuła ta jest absolutnie ciągła, więc obliczenia na takich zmiennych zależnych mogą być wykonywane z użyciem technik opisanych w poprzednim podrozdziale. Skrajne przypadki, odpowiadające kopułom M i W , wymagają osobnych metod obliczeniowych.

W pracy [82] pokazano, że jeżeli ciągle zmienne losowe X i Y są opisane łączną kopułą M , to istnieje ściśle rosnąca funkcja α taka, że $Y = \alpha(X)$. Podobnie jeżeli ciągle zmienne losowe X i Y są opisane łączną kopułą W , to istnieje ściśle malejąca funkcja β taka, że $Y = \beta(X)$. Bierze się to z faktu, że cała masa probabilistyczna w przypadku kopuł M i W jest równomiernie rozłożona na przekątnych: $v = u$ (kopuła M) oraz $v = 1 - u$ (kopuła W). Jeżeli rozkłady brzegowe są różne od jednostajnych, funkcje α i β wyznacza się z warunków:

1. W przypadku kopuły M żądamy, aby $G(y) = F(x)$, stąd dostajemy

$$y = \alpha(x) = G^{-1}(F(x)). \quad (3.16)$$

2. W przypadku kopuły W żądamy, aby $G(y) = 1 - F(x)$, stąd mamy

$$y = \beta(x) = G^{-1}(1 - F(x)). \quad (3.17)$$

F i G oznaczają, odpowiednio, dystrybuanty zmiennych X i Y .

Zmienne X i Y opisane łącznymi kopułami M lub W są powiązane zależnością deterministyczną. Rzutuje to na sposób wykonywania obliczeń na takich zmiennych. Rozważmy na przykład sumę zmiennych powiązanych zależnością M :

$$X + Y = X + \alpha(X) = \gamma(X),$$

gdzie funkcja γ jest określona następująco:

$$\gamma(x) = 1 + \alpha(x).$$

Oznacza to, że suma zmiennych losowych powiązanych zależnością opisaną kopułą M jest tak naprawdę funkcyjnym przekształceniem jednej ze zmiennych. Do obliczenia rozkładu zmiennej $\gamma(X)$ potrzebna jest oczywiście znajomość samej funkcji γ oraz funkcji do niej odwrotnej, jak i jej pochodnej (por. (2.5) i (2.6)).

Trudność w takim podejściu polega na tym, że funkcja α , a co za tym idzie, również γ zależą w dość skomplikowany sposób zarówno od danych rozkładów (brzegowych) zmiennych X i Y , jak i samej operacji wykonywanej na tych zmiennych. Automatyzując proces obliczeń, należy wyznaczyć te funkcje (α i γ) na drodze obliczeń numerycznych (lub symbolicznych) w zależności od danych rozkładów zmiennych X i Y . Dodatkową komplikacją jest konieczność odwracania funkcji γ oraz wyznaczania jej pochodnej. Ponadto w przypadku pewnych operacji funkcja γ może nie być funkcją monotoniczną, co wymaga dodatkowo wyznaczania przedziałów monotoniczności (por. (2.6)). Stosując obliczenia numeryczne oparte na wielomianach Czebyszewa opisanych w rozdz. 4, a w szczególno-

ści rozwiązywanie równań (potrzebne do wyznaczania funkcji odwrotnej) oraz różniczkowanie, proces ten można jednak efektywnie zautomatyzować. Przykłady numeryczne są przedstawione w rozdz. 6.

Warto w tym miejscu zaznaczyć, że rozważanie ogólnych zależności funkcyjnych pomiędzy dwoma zmiennymi może prowadzić do dość zaskakujących rezultatów. W pracy [82] pokazano, że dla dowolnej kopuły C można dobrać taką funkcję f , że zmienne X oraz $Y = f(X)$ będą miały kopułę łączną C_f dowolnie bliską kopule C (w normie jednostajnej).

3.5.4. Ograniczenia na dystrybuantę wyniku

Posługując się modelami kopuł W i M , można wyznaczyć najbardziej rozproszone oraz najbardziej skupione rozkłady wynikowe dla operacji na zmiennych losowych, jeżeli znamy jedynie rozkłady brzegowe tych zmiennych, natomiast nic nie wiemy na temat zależności pomiędzy nimi. Skrajne kopuły M i W są szczególnymi przypadkami kopuł osobliwych względem miary Lebesgue'a w $\mathbb{R}^2$ (cała masa probabilistyczna jest skoncentrowana na przekątnych). Rzecz jasna, istnieją bardziej skomplikowane postacie rozkładów łącznych.

Z punktu widzenia rachunku prawdopodobieństwa, aby wyznaczyć rozkład zmiennej wynikowej, potrzebna jest znajomość rozkładu łącznego. Jednak w sytuacjach praktycznych rozkład może być co najwyżej identyfikowany z pewną dokładnością. W związku z powyższym istnieje ogólny problem postawiony przez N.A. Kołmogorowa [36]: podać najlepsze możliwe ograniczenia na rozkład (dystrybuantę) sumy dwóch zmiennych losowych, jeżeli znane są jedynie rozkłady brzegowe, natomiast nie jest znany rozkład łączny obu zmiennych. Rozwiązanie tego problemu podał po raz pierwszy G.D. Markow [36, 89]. W pracy [36] udowodniono następujące twierdzenie [36, twierdzenie 5.1]:

Twierdzenie 3.4. Niech X i Y będą zmiennymi losowymi o dystrybuantach, odpowiednio, F i G , $\oplus$ ciągłą funkcją dwuargumentową (działaniem), niemalejącą ze względu na poszczególne argumenty. Wtedy zachodzą następujące ograniczenia:

$$\underline{CDF}_{X \oplus Y}(z) \leq CDF_{X \oplus Y}(z) \leq \overline{CDF}_{X \oplus Y}(z),$$

przy czym

$$\underline{CDF}_{X \oplus Y}(z) = \sup_{x \oplus y = z} W(F(x), G(y)), \quad (3.18)$$

$$\overline{CDF}_{X \oplus Y}(z) = \inf_{x \oplus y = z} \bar{W}(F(x), G(y)), \quad (3.19)$$

gdzie $\bar{W}$ oznacza kopułę dualną do kopuły W (podrozdz. 3.3.2).

Funkcje $\underline{CDF}$ i $\overline{CDF}$ są dystrybuantami pewnych rozkładów i nazywać je bę-

dziemy dystrybuantą górną i dolną ograniczającą wynik. Nietrudno zauważyć, że wyniku tego w ogólności nie da się poprawić, gdyż dla rozkładów jednopunktowych nierówności zamieniają się w równości [89, zadanie 6.1.1].

Oczywiście dla typowych rozkładów niezdegenerowanych zmiennych X i Y różnice pomiędzy $\underline{CDF}$ i $\overline{CDF}$ mogą być dość znaczne. Rozważmy na przykład dwie ciągłe oraz dodatnie zmienne losowe X i Y , natomiast interesuje nas podanie ograniczeń na sumę $X + Y$. Załóżmy dodatkowo, że $\underline{CDF}(z) < \overline{CDF}(z)$ dla $z > 0$, wtedy na podstawie ciągłości oraz własności wartości oczekiwanej [50, s. 88] mamy:

$$\begin{aligned} 0 < \int_0^\infty \overline{CDF}(z) - \underline{CDF}(z) \, dz \\ = - \int_0^\infty 1 - \underline{CDF}(z) \, dz + \int_0^\infty 1 - \overline{CDF}(z) \, dz = \underline{\mu} - \overline{\mu}, \end{aligned}$$

gdzie $\underline{\mu}$, $\overline{\mu}$ oznaczają wartości oczekiwane dystrybuant ograniczających: górnej i dolnej. Zatem w takim przypadku nie jest możliwe znalezienie rozkładów łącznych, które pozwoliłyby na osiągnięcie w wyniku obu skrajnych rozkładów $\underline{CDF}(z)$ i $\overline{CDF}(z)$. Dzieje się tak, dlatego że wartość oczekiwana sumy dwóch zmiennych losowych jest sumą wartości oczekiwanych bez względu na rodzaj zależności pomiędzy nimi. Zatem skrajne dystrybuanty $\underline{CDF}$ i $\overline{CDF}$ (ograniczenia (3.18), (3.19)) zwykle nie są możliwe do osiągnięcia jako wynik operacji $\oplus$ na zmiennych losowych.

Niemniej jednak można pokazać [89, podrozdz. 6, twierdzenie 6.1.2], że dla dowolnych zmiennych X i Y oraz dowolnych liczb $s, t \in [0, 1]$ istnieją takie kopuły C_s oraz C_t wiążące zmienne X i Y , że:

$$\begin{aligned} \underline{CDF}_{X \oplus Y}(z_0) &= CDF_{X \oplus Y}(z_0) = s, \\ \overline{CDF}_{X \oplus Y}(z_1) &= CDF_{X \oplus Y}(z_1) = t. \end{aligned}$$

Oznacza to, że ograniczeń tych ((3.18), (3.19)), przynajmniej w sensie punktowym, bez dodatkowych założeń poprawić się nie da.

Z technicznego punktu widzenia obliczenia dystrybuant (dolnej i górnej) przebiegają podobnie do obliczeń opisanych w podrozdz. 3.5. Różnica polega na tym, że zamiast całkowania należy wyznaczyć maksimum wzdłuż konturu oraz obliczenia wykonujemy na dystrybuantach łącznych zamiast na rozkładach. Operacje te w pewnym stopniu przypominają obliczenia na liczbach rozmytych [25, 62, 98]. Przykłady obliczeniowe zostaną przedstawione w rozdz. 6.

4. Metody numeryczne stosowane w obliczeniach probabilistycznych

4.1. Wstęp

Do wykonywania obliczeń na zmiennych losowych potrzebne są przede wszystkim dwie operacje numeryczne: całkowanie i interpolacja. Całkowanie pojawia się w praktycznie każdej operacji, włączając w to wykonywanie obliczeń arytmetycznych, marginalizację czy wyznaczanie rozkładów warunkowych (wzory (2.1)–(2.4)). Konieczność interpolacji jest mniej widoczna, jednak należy mieć świadomość, że całki występujące w operacjach probabilistycznych są zwykle zależne od parametrów. Nawet najprostsza operacja marginalizacji¹ wymaga zapamiętania wyniku, czyli rozkładu brzegowego, w jakiejś formie. W przypadku dyskretnym skończonym sytuacja jest prosta, ponieważ wyniki można przechować w tablicy. W przypadku ciągłym wynik należy odpowiednio interpolować lub aproksymować. Należy tu podkreślić, że wynik pojedynczej operacji musi bardzo dokładnie aproksymować prawidłowe rozwiązanie, ponieważ zwykle jest on dalej wykorzystywany w bardziej złożonych operacjach. Z tego powodu wybór odpowiedniej reprezentacji numerycznej dla zmiennych losowych jest jednym z istotnych zagadnień w obliczeniach probabilistycznych. Numeryczne operacje całkowania i interpolacji, pod warunkiem wykorzystania odpowiednich procedur, są bardzo zbliżone do siebie i wymagają podobnych nakładów obliczeniowych. Efektywność pakietu **PaCAL** opiera się przede wszystkim na odpowiednio starannej implementacji tych dwóch metod numerycznych.

Obok całkowania i interpolacji kolejną ważną operacją wykorzystywaną w obliczeniach probabilistycznych jest zamiana zmiennych. Zamiana zmiennych w ogólności wymaga rozwiązywania równań algebraicznych wielu zmiennych oraz liczenia jacobianów (podrozdz. 3.5), co w ogólności jest złożone numerycznie. Pewne prostsze operacje, jak na przykład działania arytmetyczne czy złożenia z prostszymi funkcjami, można po prostu stablicować. Takie podejście jest wykorzystywane w arytmetyce zmiennych niezależnych. W ogólnym przypadku rozwiązywania równań nie da się uniknąć. W przypadku ogólnej inferencji probabilistycznej w pakiecie **PaCAL** przy transformacjach zmiennych są wy-

¹ Marginalizacją nazywamy wyznaczenie rozkładu brzegowego jednej lub kilku zmiennych na podstawie pełnego rozkładu łącznego; operacja ta wymaga w przypadku dyskretnym sumowania względem nieinteresujących nas zmiennych, natomiast w przypadku ciągłym całkowania.

korzystywane rozwiązania symboliczne. W pewnym zakresie są stosowane również inne operacje numeryczne, jak różniczkowanie czy szukanie ekstremów.

W rozdziale tym zostaną przedstawione metody numeryczne wykorzystywane w obliczeniach probabilistycznych. Będą to przede wszystkim: interpolacja barycentryczna i aproksymacja z użyciem wielomianów Czebyszewa; metody całkowania związane z tą reprezentacją, jak całkowanie wielomianów Czebyszewa czy kwadratury Clenshawa–Curtisa; odpowiednie metody transformacji zmiennych do interpolacji i całkowania funkcji na przedziałach nieskończonych oraz funkcji zawierających osobliwości. Rozważania w tym rozdziale dotyczą operacji na funkcjach jednej zmiennej. Treść rozdziału opiera się w dużej mierze na pracach [10, 12, 79, 119, 121]. Inspiracją do wykorzystania interpolacji barycentrycznej na węzłach Czebyszewa był projekt **chebfun** napisany w programie **Matlab** [5, 120]. Projekt ten dotyczy obliczeń numerycznych z użyciem operatorów i funkcji typowych dla analizy funkcjonalnej, analogicznie do tego, jak wykonuje się operacje na macierzach i wektorach w numerycznej algebrze liniowej. Należy jednak zaznaczyć, że rozwiązania przyjęte w pakiecie **PaCAL**, zwłaszcza do interpolacji funkcji z osobliwościami oraz funkcji określonych na przedziałach nieskończonych, stanowią oryginalny wkład autorów projektu **PaCAL** [51]. Eksperymenty numeryczne przedstawione w dalszej części pokazują, że w zakresie operacji probabilistycznych rozwiązania przyjęte w projekcie **PaCAL** są konkurencyjne w stosunku do rozwiązań stosowanych w projekcie **chebfun**.

4.2. Interpolacja i aproksymacja

4.2.1. Interpolacja barycentryczna

Zarówno interpolacja, jak i aproksymacja służą do numerycznego przybliżania funkcji. Rozróżnienie pomiędzy tymi technikami polega na tym, że w zadaniu aproksymacji poszukuje się przybliżenia funkcji, kontrolując wybraną normę błędu, a w przypadku interpolacji wymaga się dodatkowo, aby wartości interpolatora² w określonych węzłach pokrywały się z wartościami oryginalnej funkcji. W takim ujęciu interpolacja jest jedną z wygodnych technik rozwiązywania zadania aproksymacji. W przypadku obliczeń probabilistycznych interesująca jest aproksymacja w normach $\|\cdot\|_1$ oraz $\|\cdot\|_\infty$. W dalszej części będziemy się posługiwać następującymi zbiorami funkcji: $C[a, b]$ – zbiór funkcji ciągłych na przedziale domkniętym $[a, b]$, $C^k[a, b]$ – zbiór funkcji k -krotnie różniczkowalnych w sposób ciągły na przedziale domkniętym $[a, b]$, $\mathcal{A}[a, b]$ – zbiór funkcji analitycznych na przedziale $[a, b]$, to jest takich, dla których istnieje szereg potęgowy zbieżny do tej funkcji.

Rozważania w tym podrozdziale dotyczą funkcji ciągłych na przedziale domkniętym i ograniczonym. Niech dana będzie funkcja $f \in C[a, b]$. Funkcja f jest próbkowana w punktach (węzłach) $x_k, k = 1, \dots, n, x_k \in [a, b]$. Wartości funkcji oznaczamy $f_k = f(x_k)$.

² Wyraz „interpolator” jest używany w znaczeniu zestawu węzłów spróbowanej funkcji wraz z ustaloną procedurą odtwarzania wartości funkcji w dowolnym punkcie między węzłami.

Niech $\{(x_k, f_k)\}_{k=1, \dots, n}$ będą różnymi węzłami interpolacji. Wielomian interpolacyjny Lagrange'a [10] stopnia n można przedstawić w postaci

$$p(x) = \sum_{k=0}^n l_k(x) f_k,$$

gdzie funkcja wagowa $l_k(x)$ ma postać:

$$l_k(x) = \frac{\prod_{j=0, \dots, n, j \neq k} (x - x_j)}{\prod_{j=0, \dots, n, j \neq k} (x_k - x_j)}.$$

Idea interpolacji barycentrycznej polega na przedstawieniu wielomianu Lagrange'a w formie barycentrycznej (ang. *barycentric formula*):

$$p_n(x) = \frac{\sum_{k=0}^n \frac{w_k}{x - x_k} f_k}{\sum_{k=0}^n \frac{w_k}{x - x_k}},$$

gdzie wagi w_k zależą jedynie od odciętych węzłów interpolacji:

$$w_k = \frac{1}{\prod_{j=0, \dots, n, j \neq k} (x_k - x_j)}.$$

Taka postać wielomianu interpolacyjnego pozwala osiągnąć znaczny zysk obliczeniowy w przypadku, gdy wagi w_k dają się łatwo wyznaczyć i stabilizować. Wtedy obliczenie wartości wielomianu interpolacyjnego w dowolnym punkcie $f(x)$ wymaga jedynie $\mathcal{O}(n)$ prostych operacji, podczas gdy wykorzystanie pełnej formuły Lagrange'a zajmuje $\mathcal{O}(n^2)$ operacji. Podobnie dodanie nowego węzła wymaga $\mathcal{O}(n)$ operacji, jeśli wagi w_k możemy wyznaczyć szybko, zamiast $\mathcal{O}(n^2)$ operacji przy bezpośrednim zastosowaniu formuły Lagrange'a. Ponadto praktyka pokazuje, że obliczenia z użyciem formuły barycentrycznej są bardzo stabilne numerycznie.

Dla węzłów szczególnej postaci znane są odpowiednie zestawy wag, niektóre z nich zostały zebrane w tab. 4.1 [119]. Z praktycznego punktu widzenia najważniejsze zestawy węzłów są pierwiastkami wielomianów ortogonalnych, jak na przykład wielomiany Czebyszewa różnego rodzaju [79, 94] czy wielomiany Legendre'a.

Wielomiany interpolacyjne są uniwersalnym aproksymatorem funkcji na przedziale domkniętym i ograniczonym. Jeżeli za reprezentację funkcji przyjmujemy interpolator, to należy wyznaczyć i stabilizować jedynie n wartości funkcji w odpowiednich węzłach. Na-

tomiast do odtworzenia funkcji konieczna jest jedynie znajomość wag oraz zastosowanie formuły barycentrycznej.

W przypadku interpolacji Newtona (na węzłach równomiernie rozłożonych) symbol Newtona powoduje zmianę wag się o kilka rzędów wielkości i komplikacje numeryczne przy obliczaniu sumy. Znane jest w związku z tym zjawisko Rungego (ang. *Runge's phenomenon*), polegające na wzroście oscylacji wielomianu interpolacyjnego w pobliżu punktów brzegowych³. Co więcej, w pracy [100] pokazano, że nawet dla bardzo regularnych funkcji analitycznych nie ma szybko zbieżnych schematów numerycznych do aproksymacji jednostajnej, opartych na równomiernym próbkowaniu funkcji. Zera wielomianów ortogonalnych mają własność zagęszczania się na końcach przedziału, co zapobiega takim oscylacjom, natomiast odpowiednie twierdzenia gwarantują wysoką jakość aproksymacji (podrozdz. 4.5).

Obliczenia z użyciem formuły barycentrycznej oparte na węzłach będących pierwiastkami wielomianów ortogonalnych są bardzo stabilne numerycznie. Warto zwrócić uwagę, że tym przypadku wagi nie zmieniają się znacznie, na przykład dla węzłów Czebyszewa drugiego rodzaju przyjmują jedynie cztery wartości: $-1, -\frac{1}{2}, \frac{1}{2}, 1$. Więcej informacji na temat interpolacji barycentrycznej można znaleźć w pracy [10].

Tab. 4.1. Przykładowe węzły x_k i wagi w_k , $k = 0, \dots, n$ dla formuły barycentrycznej na przedziale $[1, 1]$ (na podstawie [10])

Węzły	x_k	w_k
Newtona	$\frac{k}{n} + \frac{k-n}{n}$	$(-1)^k \binom{n}{k}$
Czebyszewa I rodzaju *	$\cos \frac{2k+1}{2(n+1)}\pi$	$(-1)^k \sin \frac{2k+1}{2(n+1)}\pi$
Czebyszewa II rodzaju **	$\cos \frac{k\pi}{n}$	$(-1)^k \cdot \begin{cases} \frac{1}{2}, & k=0, k=N \\ 1 & \text{dla pozostałych} \end{cases}$
Legendre'a ***	zera $L_n(x)$	$(-1)^k \sqrt{(1-x_k^2)} w_k$

* Zera wielomianów Czebyszewa drugiego rodzaju T_n .

** Rozszerzone zera wielomianów U_n – ekstrema wielomianów T_n .

*** Zera wielomianów Legendre'a stopnia n (L_n), w_k są wagami używanymi w kwadraturze Gaussa [119, rozdz. 19], rys. 4.6.

4.2.2. Wymierna interpolacja barycentryczna

W ogólnym przypadku, jeżeli wagi są dowolne, to formuła barycentryczna prowadzi do interpolacji funkcją wymierną [9]. Istnieje wiele metod doboru współczynników wagowych do uzyskania dobrej interpolacji funkcjami wymiernymi. Na przykład w pracy [34]

³ C. Runge pokazał, że funkcja $f(x) = \frac{1}{1+25x^2}$, $x \in [-1, 1]$ nie może być aproksymowana w normie $\|\cdot\|_\infty$ wielomianem interpolacyjnym z węzłami równo odległymi.

jest proponowane użycie węzłów równomiernych (podobnie jak w interpolacji Newtona) oraz wag postaci:

$$w_k^{(d)} = (-1)^k v_k^{(d)},$$

gdzie v_k jest tworzone z użyciem procedury podobnej jak przy budowie trójkąta Pascala:

$$\begin{aligned} d=0 \quad v_k &= 1 \ 1 \ 1 \ 1 \ \dots \ 1 \ 1 \ 1 \ 1, \\ d=1 \quad v_k &= 1 \ 2 \ 2 \ 1 \ \dots \ 2 \ 2 \ 2 \ 1, \\ d=2 \quad v_k &= 1 \ 3 \ 4 \ 4 \ \dots \ 4 \ 4 \ 3 \ 1, \\ d=3 \quad v_k &= 1 \ 4 \ 7 \ 8 \ \dots \ 8 \ 7 \ 4 \ 1, \\ &\dots \end{aligned}$$

Interpolacja wymierna początkowo wydawała się atrakcyjnym podejściem do interpolacji funkcji mających bieguny. Jednak taka interpolacja może być wykonana z wysoką dokładnością jedynie dla funkcji z biegunami całkowitego rzędu. Przeprowadzone próby z bardziej złożonymi funkcjami, jak na przykład $f(x) = -\log(x)$, $x \in (0, 1]$, nie wypadły zachęcająco. Znacznie lepiej w takich przypadkach działało opisanie w dalszej części rozwiązanie, polegające na transformacji zmiennych.

4.3. Obliczenia z użyciem wielomianów Czebyszewa

4.3.1. Związki z szeregiem Fouriera

Szeregi Czebyszewa są obok szeregów Fouriera jednym z podstawowych narzędzi aproksymacji. Pomiędzy tymi dwoma metodami istnieje wiele wzajemnych zależności. W tym podrozdziale zostaną przedstawione najważniejsze fakty.

Aproksymacja trygonometryczna służy do przybliżania ciągłych funkcji okresowych, lub inaczej ciągłych na okręgu jednostkowym. Wielomian trygonometryczny stopnia n można zapisać w postaci:

$$f(\varphi) = \sum_{k=-n}^n c_k \exp(ik\varphi).$$

Do określenia jednoznacznie wielomianu trygonometrycznego stopnia n wystarczy podać wartości funkcji w $2n+1$ węzłach równomiernie rozłożonych na okręgu jednostkowym (lub w przedziale $(-\pi, \pi)$) [32]: $\varphi_j = \frac{2j\pi}{2n+1}$, $j = -n, \dots, -1, 0, 1, \dots, n$. W takim przypadku wyznaczenie współczynników c_k odpowiada dyskretnemu przekształceniu Fouriera. Nie wymaga ono całkowania i sprowadza się do zastosowania wzorów Eulera–Fouriera:

$$c_k = \frac{1}{2n+1} \sum_{j=-n}^n f_j \exp(ik\varphi_j).$$

Praktycznie do wyznaczania współczynników Fouriera na podstawie wartości funkcji w węzłach służy jeden z najważniejszych algorytmów numerycznych, to jest szybka transformata Fouriera (ang. *FFT Fast Fourier Transform*).

Istnieje kilka rodzajów wielomianów Czebyszewa. Wielomiany n -tego stopnia, pierwszego rodzaju definiuje się jako:

$$T_n(x) = \cos(n\theta), \quad x = \cos \theta. \quad (4.1)$$

Podobnie definiuje się wielomiany drugiego rodzaju:

$$U_n(x) = \frac{\sin(n+1)\theta}{\sin \theta}, \quad x = \cos \theta. \quad (4.2)$$

Miejsca zerowe tych wielomianów dane są wzorami: dla wielomianu T_n

$$x_k = \cos \frac{(k - \frac{1}{2})\pi}{n}, \quad k = 1, \dots, n, \quad (4.3)$$

a dla wielomianów U_n

$$x_k = \cos \frac{k\pi}{n+1}, \quad k = 1, \dots, n. \quad (4.4)$$

Dalsze rodzaje wielomianów Czebyszewa można znaleźć w pracach [79, 94]. Pomiedzy różnymi rodzajami wielomianów Czebyszewa zachodzi szereg własności, które ułatwiają ich całkowanie, różniczkowanie oraz wykonywanie innych operacji.

Wielomiany Czebyszewa pierwszego rodzaju tworzą na przedziale $[-1, 1]$ układ ortogonalny z funkcją wagową $\frac{1}{\sqrt{1-x^2}}$, a wielomiany drugiego rodzaju tworzą podobnie układ ortogonalny z funkcją wagową $\sqrt{1-x^2}$. Sumy częściowe szeregu Czebyszewa:

$$S_n^T[f](x) = \sum_{k=0}^n a_k T_k(x)$$

są wygodnym narzędziem aproksymacji. Można pokazać, że operatory liniowe S_n [101, rozdz. 5] mają najmniejszą normę w grupie wszystkich projekcji [94, T7.6]. Nie oznacza to, że sumy częściowe dla dowolnej funkcji $f \in C[a, b]$ minimalizują normę błędu aproksymacji $\|S_n[f] - f\|_\infty$ [119]. Do znajdowania wielomianów optymalnych w normie jednostajnej służy algorytm Remeza [17, podrozdz. 3.8]. Niemniej sumy częściowe szeregu Czebyszewa dają bardzo dobre przybliżenia, niewiele gorsze od optymalnych, a – co ważniejsze – wyznaczanie współczynników Czebyszewa czy wielomianów interpolacyjnych na węzłach Czebyszewa jest znacznie szybsze.

Zwykle jako podstawowy zestaw węzłów do interpolacji wykorzystuje się rozszerzone węzły drugiego rodzaju wraz ze skrajnymi punkt dziedziny: $-1, 1$ (tab. 4.1). Z za-

leżności (4.16) oraz z własności, że zera wielomianów Czebyszewa wszystkich rodzajów są jednokrotne, wynika, że zera wielomianów Czebyszewa drugiego rodzaju U_{n-1} opowiadają wewnętrznym ekstremom wielomianów pierwszego rodzaju T_n .

Mając dane wartości funkcji f_k w rozszerzonych węzłach Czebyszewa drugiego rodzaju, można wygodnie zamieniać je na wartości współczynników Czebyszewa i odwrotnie, wykorzystując do tego celu własności szybkiej transformaty Fouriera. Współczynniki wielomianu interpolacyjnego Czebyszewa dane są wzorem [79, podrozdz. 4.6, 4.7]:

$$c_k = \sqrt{\frac{2}{n}} \sum_{j=0}^n T_k(x_j) f(x_j) = \frac{1}{2} T_k(x_0) f(x_0) + T_k(x_1) f(x_1) + \dots + \frac{1}{2} T_k(x_n) f(x_n). \quad (4.5)$$

Transformacja odwrotna, to jest wyznaczanie wartości funkcji w węzłach na podstawie znajomości współczynników $\{c_k\}$, dane jest wzorem:

$$f_j = \sqrt{\frac{2}{n}} \sum_{k=0}^n T_k(x_j) c_k = \frac{1}{2} T_0(x_j) c_0 + T_1(x_j) c_1 + \dots + \frac{1}{2} T_n(x_j) c_n. \quad (4.6)$$

Przekształcenia te zwane są niekiedy dyskretną transformatą Czebyszewa [79, podrozdz. 4.7]. Mając na uwadze, że rozszerzone węzły drugiego rodzaju (por. (4.3)) mają postać:

$$x_j = \cos \theta_j, \text{ gdzie } \theta_j = \frac{j\pi}{n}, \quad j = 0, 1, \dots, n$$

oraz fakt, że $T_k(x_j) = \cos \frac{jk\pi}{n}$ (por. (4.1)), zależność (4.5) można zapisać w postaci:

$$c_k = \sqrt{\frac{2}{n}} \sum_{j=0}^n \cos \frac{jk\pi}{n} f(\cos \theta_j), \quad k = 0, \dots, n. \quad (4.7)$$

Na podstawie wzorów Eulera ($\cos z = \frac{1}{2}(e^{iz} + e^{-iz})$) oraz parzystości funkcji $f(\cos x)$ zależność (4.7) sprowadza się do postaci bliskiej transformacji Fouriera.

$$c_k = \sqrt{\frac{1}{2n}} \sum_{j=-n}^n \exp \frac{-ijk\pi}{n} f\left(\cos \frac{j\pi}{n}\right), \quad k = -n, \dots, n. \quad (4.8)$$

Postać (4.7) nosi nazwę dyskretniej transformaty kosinusowej i dotyczy funkcji parzystych 2π -okresowych.

Geometrycznie transformacja Czebyszewa funkcji f określonej na odcinku $[-1, 1]$ odpowiada, z dokładnością do skalowania współczynników, transformacji Fouriera funkcji $g(\theta) = f(\cos \theta)$ określonej na okręgu jednostkowym (rozszerzonej parzystości na odcinek $[0, 2\pi)$). Na rysunku 4.1 przedstawiono związek węzłów trygonometrycznych (równomiernie rozłożonych na okręgu) oraz rozszerzonych węzłów Czebyszewa drugiego rodzaju jako

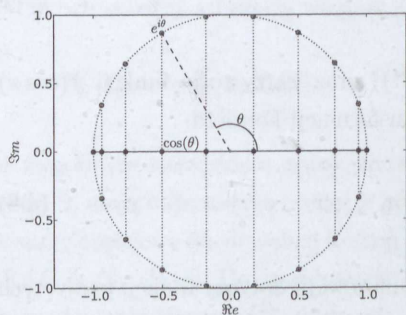
rzutów węzłów trygonometrycznych na oś rzeczywistą. Na rysunku 4.2 przedstawiono wykresy przykładowej funkcji $f(x) = \exp(x)$, $x \in [-1, 1]$ oraz przekształconej funkcji $g(\theta) = \exp(\cos \theta)$, $\theta \in [0, 2\pi]$. Jak widać, funkcja g jest gładką funkcją 2π -okresową. Poniższy fragment kodu w języku Python wykonuje dyskretną transformatę Czebyszewa, wykorzystując szybką transformatę Fouriera.

```
1 def chebtransform(fk):
2     n = len(fk)
3     oncircle = hstack((fk[-1::-1], fk[1:-1]))
4     fftcoef = real(fft(onscircle)) / (2 * n - 2)
5     fftcoef[n-1:0:-1] += fftcoef[n-1:]
6     return fftcoef[n-1::-1]
```

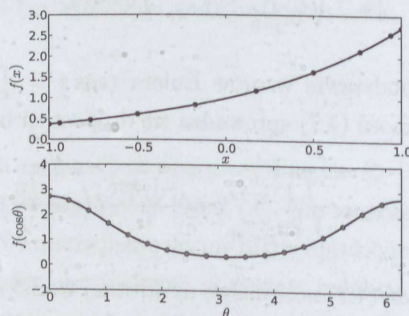
Natomiast przekształcenie odwrotne wykonuje kolejna funkcja.

```
1 def ichhebtransform(ck):
2     n = len(ck)
3     oncircle = hstack((ck[-1], ck[-2:0:-1]/2, ck[0:-1]/2))
4     v = real(ifft(onscircle))
5     f=(n-1)*hstack(( [2*v[0]], v[1:n-1]+v[-1:n-1:-1], [2*v[n-1]] ))
6     return f
```

Zamiana sposobu reprezentacji funkcji pomiędzy węzłami interpolacyjnymi a współczynnikami Czebyszewa jest wygodna z kilku powodów. Przede wszystkim do wyznaczania wartości funkcji trochę wygodniejsze jest użycie węzłów interpolacyjnych oraz wykorzystanie formuły barycentrycznej. Obliczanie sumy częściowej szeregu Czebyszewa jest



Rys. 4.1. Związek pomiędzy węzłami interpolacji trygonometrycznej (na okręgu) a węzłami interpolacji wielomianami Czebyszewa drugiego rodzaju



Rys. 4.2. Wykorzystanie dyskretniej transformacji Fouriera funkcji $f(\cos \theta)$ do wyznaczania współczynników wielomianu interpolacyjnego Czebyszewa funkcji $f(x)$

nieco bardziej skomplikowane. Zwykle wykorzystuje się algorytm Clenshawa [79, podrozdz. 2.4.1]. Jednak oba podejścia wymagają podobnej liczby $\mathcal{O}(n)$ prostych operacji arytmetycznych do wyznaczenia wartości w pojedynczym punkcie. Do wyznaczenia wartości w wielu szczególnych punktach, to jest węzłach, wykorzystuje się pokazaną wyżej transformację Czebyszewa, która dzięki zastosowaniu szybkiej transformaty Fouriera wymaga $\mathcal{O}(n \log n)$ operacji arytmetycznych. Rozwinięcia Czebyszewa są wygodne do wykonywania wielu operacji numerycznych, jak: różniczkowanie, całkowanie czy wyznaczanie pierwiastków.

4.3.2. Przyrostowy algorytm interpolacji

Węzły interpolacyjne Czebyszewa drugiego rodzaju zagnieżdżają się w sobie. Oznacza to, że jeżeli mamy spróbkowaną funkcję w n węzłach Czebyszewa, to do wyznaczenia interpolatora w $2n - 1$ węzłach wystarczy obliczyć funkcję w dodatkowych $n - 1$ węzłach umiejscowionych pomiędzy początkowymi n węzłami. Węzły Czebyszewa pierwszego rodzaju również zagnieżdżają się w sobie, jednak w bardziej skomplikowany sposób⁴. Mając n początkowych węzłów, możemy wyznaczyć interpolator, wykorzystujący $3n$ węzłów wstawionych po dwa węzły pomiędzy każdą kolejną parę węzłów oraz po jednym węzle po bokach. W ten sposób do budowy nowego interpolatora na n węzłach wykorzystujemy jedną trzecią węzłów wyznaczonych w poprzednim kroku, a musimy obliczyć wartości funkcji w kolejnych dwóch trzecich. Ilustracja zagnieżdżania się węzłów Czebyszewa została przedstawiona na rys. 4.3 i 4.4.

Sposób zagnieżdżania się węzłów sugeruje wykorzystanie algorytmu adaptacyjnego do konstrukcji interpolatora ustalonej funkcji (alg. 1). Taki sposób konstrukcji interpolatora zbudowanego na n węzłach, połączony z kontrolą błędu, wymaga jedynie $2n + 1$ -krotnego (lub $3n$ w przypadku węzłów pierwszego rodzaju) obliczenia wartości funkcji.

Z praktycznego punktu widzenia trudno jest jednoznacznie powiedzieć, który rodzaj węzłów jest najlepszy. Na przykład w pracy [119] pokazano, że jakość interpolacji na węzłach Legendre'a jest porównywalna z węzłami Czebyszewa. Podobnie ma się sprawa z innymi wielomianami ortogonalnymi z tym wyjątkiem, że nie w każdym przypadku dysponujemy odpowiednimi zestawami wag do formuły barycentrycznej. Za węzłami Cze-

⁴ Wydaje się, że ten sposób zagnieżdżania się węzłów Czebyszewa pierwszego rodzaju nie został do tej pory zauważony. Nie ma wzmianki o tym w pracach [5, 10], ponadto w kodzie pakietu **chebfun** takie zagnieżdżanie nie jest stosowane, mimo że jest możliwe użycie węzłów pierwszego rodzaju (por. funkcja `@fun/private/growfun.m` linie 73, 74). Zagnieżdżanie się węzłów drugiego rodzaju jest opisane oraz wykorzystywane, chociaż w plikach pomocy jest wzmianka, że takie zagnieżdżanie nie daje spodziewanego zysku (por. `help chebfunpref` opcja `resampling`). Dzieje się tak prawdopodobnie dlatego, że zwykle wartości funkcji są wyznaczane stosunkowo szybko, porównywalnie do kopiowania elementów macierzy, jednak dla bardziej złożonych funkcji, których czas obliczania jest znaczny (na przykład funkcje dane za pomocą całek, jak `splot`), takie przyrostowe wykorzystanie poprzednio obliczonych węzłów daje duży zysk.

funkcji interpolacja na węzłach wewnętrznych jest znacznie bardziej atrakcyjna. Przejście pomiędzy bazami T_n i U_n najwygodniej jest realizować, przepróbkując funkcje w odpowiednich węzłach.

Aproksymowaną funkcję f można reprezentować na dwa sposoby:

- jako listę wartości funkcji $(f_0, \dots, f_n)$ w węzłach Czebyszewa odpowiedniego rodzaju;
- jako listę współczynników występujących w n -tej sumie częściowej szeregu Czebyszewa pierwszego rodzaju:

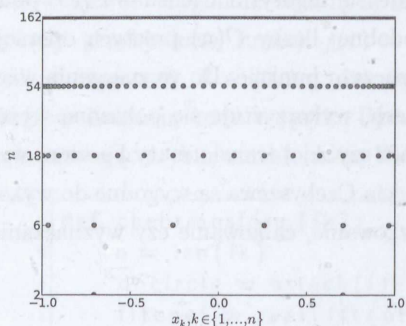
$$S_n^T[f](x) \approx \sum_{k=0}^n a_k T_k(x) = \frac{1}{2} a_0 T_0(x) + a_1 T_1(x) + \dots + a_n T_n(x) \quad (4.9)$$

lub drugiego rodzaju:

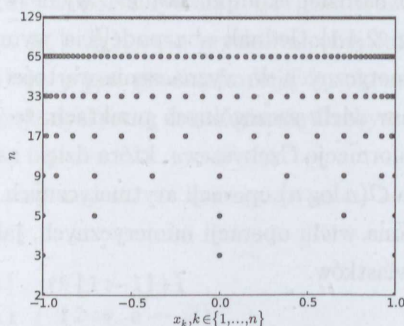
$$S_n^U[f](x) \approx \sum_{k=0}^n b_k U_k(x). \quad (4.10)$$

Jak wspomniano, wygodniej jest odtwarzać aproksymowaną funkcję z użyciem listy wartości funkcji określonych w węzłach Czebyszewa oraz formuły barycentrycznej, zamiast posługiwać się bezpośrednio sumą częściową szeregu Czebyszewa i listą współczynników. Jest to zwłaszcza wygodne przy budowie interpolatora w sposób adaptacyjny, z wykorzystaniem zagnieżdżania się węzłów (alg. 1). Przejście pomiędzy tymi dwoma rodzajami reprezentacji ustala transformacja Czebyszewa, a praktycznie szybka transformacja Fouriera (prosta i odwrotna). Do samego odtwarzania funkcji znajdowanie współczynników Czebyszewa nie jest konieczne, jednak pewne operacje numeryczne wygodniej jest wykonywać bezpośrednio na sumie częściowej szeregu Czebyszewa, korzystając z własności wielomianów Czebyszewa. Pakiet **PaCAL**, podobnie jak **chebfun**, opiera się w znacznym stopniu na interpolacji barycentrycznej. W nowszych wersjach biblioteki **numpy** języka Python jest dostępny moduł **polynomial.chebyshev** implementujący alternatywne podejście wykorzystujące rozwinięcia w szereg Czebyszewa⁶. Zaimplementowano w nim podstawowe operacje na szeregach Czebyszewa, jak całkowanie, różniczkowanie, algorytm Clenshawa czy operacje arytmetyczne na szeregach Czebyszewa. Na uwagę zasługuje fakt, że moduł ten nie ma możliwości bezpośredniego rozwinięcia w szereg Czebyszewa na podstawie węzłów Czebyszewa oraz metody FFT. Dostępna jest jedynie funkcja umożliwiająca dopa-

⁶ Moduł ten został dodany do biblioteki **numpy**, począwszy od wersji 1.4., która ukazała się 27.06.2010 r. Wszystkie operacje w tym module dotyczą funkcji na przedziałach ograniczonych (w zasadzie na przedziale $[-1, 1]$). Począwszy od wersji 1.6.0, która pojawiła się 15.05.2011 r., dostępne są analogiczne operacje na wielomianach Hermite'a i Laguerre'a. Rozpoczęcie prac na projektem **PaCAL** nastąpiło 20.01.2009 r., a pierwsza wersja (v0.99) biblioteki **PaCAL** została opublikowana 20.02.2011 r., natomiast pierwszą wersję pakietu **chebfun** (niezawierającą jeszcze funkcji na przedziałach nieskończonych) opublikowano w 2004 r. Projekt **PaCAL** jest niezależny od modułu **polynomial.chebyshev**.



Rys. 4.3. Zagnieżdżanie się węzłów Czebyszewa pierwszego rodzaju, reguła:
 $n_{i+1} = n_i + 2 \cdot n_i$



Rys. 4.4. Zagnieżdżanie się węzłów Czebyszewa drugiego rodzaju, reguła
 $n_{i+1} = n_i + (n_i - 1)$

Alg. 1. Przyrostowy algorytm konstrukcji interpolatora wykorzystujący zagnieżdżanie się węzłów Czebyszewa.

```

function  $S_n = \text{INCREMENTALINTERPOLATOR}(f, \text{absTol})$ 
   $n := 2$ 
   $x := \text{init\_nodes}(n)$                                 ▷ inicjacja węzłów, wzory (4.3), (4.4)
   $y := \emptyset$ 
  repeat
     $x := x \cup y$ 
     $S_n[f] := \text{interpolator}(f, x)$                       ▷ interpolator funkcji  $f$  na węzłach  $x$ 
     $n := n + (n - 1)$                                     ▷ lub  $n := n + 2n$  dla wielomianów I rodzaju
     $y := \text{incremental\_nodes}(n)$                         ▷ dodawane przyrostowo węzły, rys. 4.3, 4.4
     $\text{err}_{\max} := \max_y \{f(y) - S_n[f](y)\}$              ▷ szacowanie dokładności
  until  $\text{err}_{\max} < \text{absTol}$ 
  return  $S_n$ 
end function

```

byszewa przemawia możliwość skorzystania z szybkiej transformaty Fouriera [79, 119] oraz ogólne teoretyczne fakty dotyczące aproksymacji wielomianami Czebyszewa [17, 94, 101]. Wybór pomiędzy węzłami Czebyszewa pierwszego i drugiego rodzaju również nie jest jednoznaczny. Istnienie węzłów na końcach przedziałów pomaga trochę lepiej kontrolować zachowanie funkcji na brzegach przedziałów, co ma znaczenie przy sklejaniu funkcji (rozdz. 2), ponadto reguła zagnieżdżania się węzłów ($2n - 1$ zamiast $3n$) jest w tym przypadku trochę bardziej oszczędna. Z drugiej strony w obliczeniach często pojawiają się funkcje niemające dobrze określonych wartości na końcach przedziałów⁵. Dla takich

⁵ Na przykład funkcja $f(x) = \frac{x^2}{x}$ nie jest określona w zerze, mimo że poza zerem jest to bardzo regularna funkcja liniowa, natomiast w zerze ma osobliwość usuwalną. Tego typu osobliwości, usuwalne, jak i nieusuwalne, pojawiają często się przy całkowaniu wyrażeń probabilistycznych (zależności (2.3) czy (2.4)).

sowanie sum częściowych szeregu Czebyszewa do danych, analogiczne do funkcji polyfit (dostępnej zarówno w Matlab, jak i w **numpy**).

4.3.3. Całkowanie szeregów Czebyszewa

Przy wyznaczaniu całek nieoznaczonych wykorzystujemy tożsamości:

$$\int T_n dx = C + \begin{cases} \frac{1}{2} \left(\frac{T_{n+1}(x)}{n+1} - \frac{T_{n-1}(x)}{n-1} \right), & n > 1, \\ \frac{1}{4} T_2(x), & n = 1 \end{cases} \quad (4.11)$$

lub w przypadku wielomianów drugiego rodzaju:

$$\int U_n dx = \frac{1}{n+1} T_{n+1} + C, \quad (4.12)$$

gdzie C jest stałą całkowania. Druga postać wydaje się nieco wygodniejsza, jednak wymaga zmiany bazy wielomianów.

Z tożsamości tych korzysta się w ten sposób, że całkuje się odpowiednie sumy częściowe (4.9) wyraz za wyrazem, a następnie porządkuje się współczynniki, otrzymując odpowiednio:

$$\int f(x) dx \approx I_n^T[f](x) = \int S_n^T[f](x) dx = \sum_{k=0}^{n+1} \frac{a_{k-1} - a_{k+1}}{2k} T_k(x) + C, \quad (4.13)$$

gdzie $a_{-1} = a_{n+1} = a_{n+2} = 0$, oraz

$$\int f(x) dx \approx I_n^U[f](x) = \int S_n^U[f](x) dx = \sum_{k=0}^n \frac{b_k}{k} T_k(x) + C. \quad (4.14)$$

Zależności te można wykorzystać do liczenia zarówno całek oznaczonych, jak i całek skumulowanych [79, rozdz. 8].

4.3.4. Różniczkowanie szeregów Czebyszewa

Bezpośrednie różniczkowanie formuły barycentrycznej prowadzi do wyrażenia, które nie jest formułą barycentryczną, co powoduje komplikacje przy reprezentacji wyniku, jak i przy liczeniu pochodnych wyższych rzędów. Wzory na różniczkowanie funkcji w postaci formuły barycentrycznej podano w pracy [9]. Natomiast pochodne interpolowanych funkcji, podobnie jak całki, wygodnie jest wyznaczać na podstawie rozwinięć Czebyszewa. Wykorzystuje się tu tożsamości:

$$\frac{d}{dx} T_n(x) = 2n \sum_{\substack{k=0 \\ 2|n-k+1}}^n T_k(x) \quad (4.15)$$

oraz

$$\frac{d}{dx} T_n(x) = nU_{n-1}(x), \quad n = 1, \dots \quad (4.16)$$

Różniczkując sumy częściowe szeregów (4.9) i (4.10) wyraz za wyrazem oraz wykorzystując zależności (4.15) i (4.16), dostajemy przybliżenia na pochodne funkcji:

$$f'(x) \approx D_n^T[f](x) = (S_n^T[f](x))' = \sum_{k=0}^n{}' A_k T_k(x), \quad (4.17)$$

gdzie $A_k = \sum_{\substack{j=k+1 \\ 2|n-k+1}}^{n+1} 2ka_k$, oraz

$$f'(x) \approx D_n^U[f](x) = (S_{T_n}[f](x))' = \sum_{k=0}^n{}' a_{k+1} U_k(x). \quad (4.18)$$

Tożsamości (4.11) i (4.16) można również wykorzystać do zamiany rozwinięcia w szeregi Czebyszewa względem bazy wielomianów pierwszego i drugiego stopnia. Zamianę interpolatora wykonuje się, bezpośrednio wyznaczając wartości interpolowanej funkcji w odpowiednich węzłach.

Podobne rozwinięcia Czebyszewa jak dla całek i pochodnych można podać dla sumy, iloczynu i ilorazu oraz innych operacji na funkcjach zapisanych z użyciem sum częściowych szeregu Czebyszewa [79, 94]. Należy pamiętać, że zarówno całkowanie, jak i różniczkowanie sum częściowych można wykonywać w każdej sytuacji, jednak zbieżność pochodnych i całek sum częściowych do pochodnych i całek funkcji oryginalnej zachodzi przy pewnych założeniach [31, podrozdz. XII.2]. Ponadto w ogólności dla uzyskania numerycznej zbieżności całki czy pochodnej może być wymaganych więcej wyrazów rozwinięcia, niż wynika to z rozwinięcia funkcji oryginalnej.

4.3.5. Miejsca zerowe, ekstrema

Do wyznaczania zer wielomianu zapisanego w bazie jednomianów

$$w(x) = a_0 + a_1x^1 + \dots + a_nx^n \quad (4.19)$$

można wykorzystać macierz stowarzyszoną Frobeniusa (ang. *Frobenius companion matrix*):

$$C(w) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -\frac{c_0}{c_n} & -\frac{c_1}{c_n} & -\frac{c_2}{c_n} & \cdots & -\frac{c_{n-1}}{c_n} \end{bmatrix}. \quad (4.20)$$

Dowodzi się, że zera wielomianu $w(x)$ oraz wartości własne macierzy $C(w)$ są takie same. W pracach [13, 23] pokazano, w jaki sposób można skonstruować macierz stowarzyszoną dla wielomianów zapisanych w dowolnej bazie ortogonalnej. Wynika stąd na przykład, że dla wielomianu zapisanego w bazie składającej się wielomianów Czebyszewa drugiego rodzaju (4.9) stowarzyszona macierz typu Frobeniusa ma postać:

$$C^T(S_N^T) = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & \cdots & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \ddots & \ddots & \frac{1}{2} \\ \left(-\frac{a_0}{2a_n}\right) & \left(-\frac{a_1}{2a_n}\right) & \left(-\frac{a_2}{2a_n}\right) & \cdots & \left(\frac{a_{n-2}}{2a_n} + \frac{1}{2}\right) & \left(-\frac{a_{n-1}}{2a_n}\right) \end{bmatrix}, \quad (4.21)$$

lub inaczej $C^T(S_N^T) = [C_{i,j}]_{n \times n}$, gdzie

$$C_{i,j}^T = \begin{cases} 1, & i = 1, j = 2, \\ \frac{1}{2}, & i = 2, \dots, n-1, j = i-1, i+1, \\ -\frac{a_{j-1}}{2a_n}, & i = n, j = 1, \dots, n, \\ -\frac{a_{n-1}}{2a_n} + \frac{1}{2}, & i = n, j = n-1, \\ 0, & \text{w pozostałych przypadkach.} \end{cases} \quad (4.22)$$

Do wyznaczenia pierwiastków wielomianu zapisanego w bazie wielomianów T_i wystarczy znaleźć wartości własne macierzy C^T . Bardziej wydajny algorytm obliczania pierwiastków dla dużych wartości n ($n > 100$) opisano pracy [13].

Z punktu widzenia obliczeń na zmiennych losowych znajdowanie pierwiastków nie jest operacją krytyczną, jednak ma ona pewne znaczenie dla znajdowania maksimów rozkładów czy punktów przegięcia oraz przy rozwiązywaniu równań. Wyznaczanie ekstremów odbywa się w sposób naturalny – liczy się pochodną oraz jej miejsca zerowe. Takie podejście jest znacznie lepsze od zastosowania dowolnej metody optymalizacji, ponieważ

jedną metodą dostajemy wszystkie potencjalne punkty, w których mogą być ekstrema⁷. Ponadto dokładność numeryczna takich rozwiązań jest zwykle bliska precyzji obliczeń zmiennopozycyjnych.

4.4. Całkowanie numeryczne

4.4.1. Całki na przedziale skończonym

Zadanie całkowania numerycznego jest w ogólności problemem trudnym. Można pokazać [38], że dla każdej procedury numerycznej można wskazać całkowaną funkcję ciągłą, taką że dana metoda całkowania będzie dla tej funkcji rozbieżna. Pośrednie uzasadnienie takiego stanu rzeczy widać w istniejących bibliotekach numerycznych, które zwykle dostarczają po kilka metod całkowania dla funkcji różnego typu, przerzucając w ten sposób na użytkownika konieczność wyboru (por. np. `>> help quad` w programie Matlab czy biblioteki **GSL** [37] i **QUADPACK** [99]). Z punktu widzenia obliczeń probabilistycznych całkowanie jest najbardziej podstawową operacją i jakość tej operacji rzutuje na jakość całości obliczeń.

Istnieją dwa ogólne podejścia numeryczne do liczenia całek: metody adaptacyjne oraz kwadratury typu Gaussa. Procedury adaptacyjne polegają na adaptacyjnym dzieleniu przedziału całkowania na mniejsze przedziały, w których całki zwykle liczy się ustaloną prostą metodą do momentu, aż zostanie zaobserwowana zbieżność. Procedury adaptacyjne są zwykle lepiej dostosowane do radzenia sobie z dowolnymi funkcjami (nieciągłymi czy nieróżniczkowalnymi). Przegląd stosowanych praktycznie metod adaptacyjnych można znaleźć w pracy [38]. Proste procedury adaptacyjne zwykle nie osiągają tak wysokiej dokładności, zwłaszcza dla funkcji regularnych, jak kwadratury oparte na metodzie Gaussa. Kwadratury Gaussa to wspólna grupa metod, w których idea całkowania polega na przedstawieniu całki w postaci kombinacji liniowej wartości funkcji wyznaczonych w odpowiednio dobranych węzłach:

$$\int_a^b f(x) dx = \sum_{k=0}^n w_k f(x_k).$$

Węzły wybiera się jako miejsca zerowe wielomianów ortogonalnych, a wagi wyznacza się z warunku, aby metoda dawała dokładne wyniki dla wszystkich wielomianów odpowiedniego stopnia. Klasyczna kwadratura Gaussa jest oparta na wielomianach Legendre'a, punkty x_k są zerami tych wielomianów. Odpowiednikiem procedury Gaussa wykorzystującym wielomiany Czebyszewa jest kwadratura Gaussa–Czebyszewa [79, podrozdz. 8.2]. Podobną metodą polegającą na aproksymacji szeregiem Czebyszewa jest metoda Clenshawa–Curtisa [79, rozdz. 8]. Wykorzystuje ona rozszerzone węzły Czebyszewa drugiego rodzaju. Implementacje kwadratur Gaussa i Clenshawa–Curtisa w języku Pyh-

⁷ Należy pamiętać, że rozważania dotyczą funkcji jednej zmiennej.

ton przedstawiono na rys. 4.5 i 4.6 (na podstawie [121]). W pracy [121] przedstawiono szczegółowe porównanie obu metod. Istnieją warianty metody Clenshawa–Curtisa, podane historycznie znacznie wcześniej przez L. Fejéra [27], które wykorzystują węzły pierwszego rodzaju (pierwsza reguła Fejéra) oraz węzły drugiego rodzaju bez punktów brzegowych (druga reguła Fejéra). Pewne szczegóły implementacyjne metod Clenshawa–Curtisa i Fejéra omówiono w pracy [123].

Bardziej szczegółowy przegląd metod całkowania numerycznego można znaleźć w monografii [22]. W projekcie **PaCAL** funkcje są reprezentowane z wykorzystaniem funkcji kawałkami gładkich, a punkty potencjalnie osobiwie są kontrolowane w trakcie obliczeń. Z tego powodu nie ma potrzeby stosowania metod adaptacyjnych. Jako podstawowy integrator została wybrana metoda Fejéra drugiego rodzaju (nazywana we współczesnej terminologii metodą Clenshawa–Curtisa bez punktów na brzegach). Brak punktów brzegowych ułatwia całkowanie funkcji źle określonych na końcach przedziału całkowania, co jest częstym przypadkiem w obliczeniach na zmiennych losowych. Metoda ta sprawowała się znacznie lepiej od istniejących integratorów referencyjnych, dostępnych w różnych pakie-

```

1 def clenshawcurtis(f, n):
2     x = cos(pi * arange(n + 1) / n)
3     fx = f(x) / (2 * n);
4     g = real(fft(fx[range(n + 1) + range(n - 1, 0, -1)]))
5     a = g[range(0, n + 1)]
6     a[1:-1] += g[2 * n - 1: n:-1]
7     w = zeros_like(a)
8     w[:,2] = 2.0 / (1 - arange(0, n + 1, 2) ** 2)
9     l = dot(w, a);
10    return l

```

Rys. 4.5. $n + 1$ -punktowa kwadratura Clenshawa–Curtisa na przedziale $[-1, 1]$ w języku Python

```

1 def gaussquad(f, n):
2     b = 0.5 / sqrt(1 - arange(2, 2 * n + 1, 2) ** -2.0)
3     T = diag(b, 1) + diag(b, -1)
4     x, V = eig(T)
5     w = 2.0 * V[0] ** 2
6     l = dot(w, f(x))
7     return l

```

Rys. 4.6. $n + 1$ -punktowa kwadratura Gaussa na przedziale $[-1, 1]$ w języku Python

tach obliczeń numerycznych, jak **scipy.integrate** języka Python czy procedury programu Matlab, na co przykłady znajdują się w dalszej części rozdziału.

4.4.2. Przedziały nieskończone, funkcje z osobliwościami

Wszystkie przedstawione wcześniej metody interpolacji i całkowania dotyczą w zasadzie funkcji gładkich na przedziałach domkniętych. Zarówno całkowanie, jak i interpolacja funkcji na przedziałach nieskończonych (półnieskończonych) wymaga zwykle dodatkowych zabiegów technicznych. W tym przypadku należy mieć świadomość, że zbieżność jednostajna (w normie $\|\cdot\|_\infty$) w zbiorze $C[a, b]$ pociąga za sobą zbieżność w normie $\|\cdot\|_1$). Wynika to z twierdzenia Weierstrassa o osiąganiu kresów oraz jednorodności całki:

$$\int_a^b |f_n(t) - f(t)| dt \leq \int_a^b \epsilon dt \leq \epsilon(b-a).$$

Niestety, na przedziale nieskończonym tak być nie musi. Na przykład biorąc $f(t) = 0$ oraz $f_n(t) = \frac{1}{n}$ dla $t \in [-\infty, \infty]$, mamy, że f_n zbiega jednostajnie do f . Natomiast dla każdego n całki $\int_{-\infty}^{\infty} 1/n dt$ są rozbieżne. Co więcej, przykład ten można zmodyfikować, tak aby dla każdego n norma $\|f_n\|_1$ była ograniczona. Zwykle bez dodatkowych założeń na temat całkowanej funkcji nie można podać ogólnej zbieżnej procedury obliczania całek niewłaściwych dla wszystkich funkcji całkowalnych.

W tym miejscu widać, że nawet zapewniając aproksymację jednostajną funkcji na przedziale nieskończonym, trudno jest zagwarantować aproksymację w normie $\|\cdot\|_1$. Uwaga ta ma duże znaczenie dla obliczeń na zmiennych losowych, ponieważ zwykle dopiero całki z funkcji gęstości (jak dystrybuanta, mediana, średnia) mają znaczenie praktyczne. Druga uwaga jest taka, że w przypadku funkcji szybko znikających w nieskończoności, jak na przykład funkcja gęstości rozkładu normalnego, można łatwo wybrać stosunkowo wąski przedział skończony, w którym znajduje się cała masa rozkładu. Natomiast dla rozkładów o tak zwanych ciężkich ogonach (ang. *heavy-tailed distributions*) jest to trudne do zrobienia, gdyż przedział taki może być dość szeroki. Na przykład dla rozkładu normalnego symetryczny przedział ufności zawierający $1 - 1 \times 10^{-10} = 0.9999999999$ masy probabilistycznej jest równy $(-6.4669, 6.4669)$, taki sam przedział ufności dla rozkładu Cauchy'ego wynosi $(-6.3661 \times 10^9, 6.3661 \times 10^9)$, a dla rozkładu Levy'ego przedział ten jeszcze szerszy i wynosi $(0.0231, 2.5464 \times 10^{20})$. Z tych powodów zapewnienie odpowiedniej aproksymacji oraz całkowania w takiej ogólności jest dość trudne numerycznie.

Całki na przedziale nieskończonym

Procedury całkowania numerycznego na przedziale nieskończonym można podzielić na trzy grupy: metody adaptacyjne (w tym przejście do granicy), kwadratury na przedziałach nieskończonych oraz sprowadzanie do całkowania na przedziale skończonym przez zamianę zmiennych.

Metody adaptacyjne tak naprawdę wykonują całkowanie po przedziale skończonym w taki sposób, że jeden lub oba końce się poszerzają (przejście do granicy) aż do uzyskania zbieżności całki. Metody te generalnie działają dość dobrze dla funkcji szybko znikających w nieskończoności oraz nieco gorzej dla funkcji wolno zbieżnych. Poszerzanie przedziału całkowania może być stosunkowo łatwo wykonane w przypadku rozkładu normalnego, jednak dla rozkładów z ciężkimi ogonami końce takich pod przedziałów mogą różnić się o kilka rzędów wielkości.

Kwadratury typu Gaussa na przedziałach nieskończonych wykorzystują wielomiany ortogonalne z odpowiednio dobraną funkcją wagową. Często są tu stosowane wielomiany Laguerre'a, które są ortogonalne z wagą $w(x) = \exp(-x)$ na przedziale $[0, \infty)$, czy wielomiany Hermite'a ortogonalne z wagą $w(x) = \exp(-x^2)$ na przedziale $(-\infty, \infty)$ [22, podrozdz. 3.6]. Taki wybór wag ponownie powoduje, że metody te lepiej działają dla funkcji szybko znikających w nieskończoności.

Trzecia grupa metod wykorzystuje odpowiednie podstawienia, tak aby całkę na przedziale nieskończonym zastąpić całką właściwą na przedziale skończonym. Problemem jest w tym przypadku dobór przekształcenia. Idea polega na wykorzystaniu twierdzenia o zamianie zmiennych pod znakiem całki:

$$\int_a^b f(x) dx = \int_{\varphi^{-1}(a)}^{\varphi^{-1}(b)} f(\varphi(t)) \varphi'(t) dt.$$

Jeżeli funkcja φ spełnia założenia twierdzenia, $\varphi^{-1}(a)$ i $\varphi^{-1}(b)$ są granicami właściwymi, a funkcja podcałkowa $f(\varphi(t))\varphi'(t)$ jest ograniczona, to całkowanie na przedziale nieskończonym sprowadza się do obliczenia całki właściwej. W takim przypadku można użyć ogólnych metod całkowania z poprzedniego podrozdziału.

Wybór odpowiedniego podstawienia pozostaje problemem. Jeżeli założymy, że funkcja określona na przedziale półnieskończonym spełnia warunek $f = \mathcal{O}(1/x^\beta)$ dla $x \rightarrow \infty$ i $\beta > 1$, to zamiana zmiennej $t = x^\alpha$ dla $\alpha > 1/(\beta - 1)$ prowadzi do całki właściwej względem t .

Całki niewłaściwe na przedziałach skończonych

Niektóre algorytmy obliczania całek właściwych mogą być bezpośrednio zastosowane do obliczania całek niewłaściwych na przedziałach skończonych [12, 14, 22, rozdz. 16]. Umożliwiają to zwykle te algorytmy, które nie korzystają z wartości funkcji na końcach przedziałów. Takie podejście jednak ma zwykle słabą dokładność i generalnie całki niewłaściwe na przedziałach skończonych wymagają takich samych metod jak całki na przedziale nieskończonym.

Funkcję mającą osobliwość w zerze spełniającą warunek $f = \mathcal{O}(1/x^\beta)$ dla $x \rightarrow 0$, $0 < \beta < 1$ za pomocą podstawienia $t = x^\alpha$ dla $\alpha > 1/(1 - \beta)$ można sprowadzić do całki właściwej [22, podrozdz. 2.13, 3.1].

4.5. Dokładność aproksymacji

4.5.1. Rezultaty teoretyczne

Podstawowe twierdzenie Weierstrassa o aproksymacji jednostajnej mówi, że dla dowolnej funkcji ciągłej określonej na przedziale domkniętym i ograniczonym (zwartym) istnieje ciąg wielomianów zbieżny do tej funkcji w normie $\|\cdot\|_\infty$. Rozluźnienie założeń (rezygnacja z ograniczoności lub domkniętości dziedziny) w tym twierdzeniu nie jest możliwe, na co są znane stosowne kontrprzykłady⁸. Twierdzenie aproksymacyjne Weierstrassa jest zbyt ogólne, aby można było na jego podstawie szacować dokładność aproksymacji. W praktyce korzysta się z przybliżeń sumami częściowymi szeregów ortogonalnych (zwanych projekcjami⁹), jak szeregi Fouriera czy Czebyszewa. Niestety, w klasie projekcji nie można zapewnić zbieżności aproksymacji dla dowolnych funkcji ciągłych na przedziale domkniętym i ograniczonym [32, 94, 101]. Zbieżność szeregów ortogonalnych można w niektórych przypadkach w pewnym stopniu poprawić. Na przykład dla szeregów Fouriera można zastosować technikę sumacyjną Fejéra, biorąc średnie arytmetyczne sum częściowych [32, 101]. Dostajemy wtedy zbieżność jednostajną tego typu wielomianów trygonometrycznych dla dowolnych funkcji ze zbioru $C[a, b]$, jednak dzieje się to kosztem tempa zbieżności dla bardziej regularnych funkcji. Z tego powodu na aproksymowane funkcje nakłada się zwykle dodatkowe warunki, jak różniczkowalność czy warunek Lipschitza [32, 94].

Z praktycznego punktu widzenia interesująca jest jakość aproksymacji z użyciem sum częściowych szeregów Czebyszewa lub, co na jedno wychodzi, interpolacji barycentrycznej na węzłach Czebyszewa. W obliczeniach probabilistycznych ważne jest utrzymanie zarówno zbieżności jednostajnej (w normie $\|\cdot\|_\infty$), jak i zbieżności w normie $\|\cdot\|_1$.

Dla odpowiednio regularnych funkcji ciągłych można posługiwać się sumami częściowymi szeregów Czebyszewa, mając gwarancję na błąd aproksymacji. Stanowi o tym następujące twierdzenie [79, twierdzenie 5.14]:

Twierdzenie 4.1. Niech $f \in C^{k+1}[-1, 1]$, wtedy ciąg sum częściowych szeregu Czebyszewa $S_n[f]$ aproksymuje funkcję f i prawdziwe jest oszacowanie błędu:

$$\|f - S_n[f]\|_\infty \leq \mathcal{O}(n^{-k}).$$

Jeżeli f jest analityczna w otoczeniu przedziału $[-1, 1]$, to

$$\|f - S_n[f]\|_\infty \leq \mathcal{O}(r^{-n}) \text{ dla pewnego } r < 1.$$

⁸ Na przykład nie można aproksymować ciągiem wielomianów funkcji: $f(x) = \exp(x)$ dla $x \in (-\infty, +\infty)$ oraz $f(x) = \frac{1}{x}$ dla $x \in (0, 1)$.

⁹ Ustalmy podprzestrzeń liniową $L_n \subset C[a, b]$; projekcją nazywamy operator liniowy $l \in \text{Lin}(C[a, b], L_n)$ taki, że dla dowolnej $u \in L_n$ zachodzi $l(u) = u$ [101, rozdz. 4].

Wersję tego twierdzenia przy słabszych założeniach można znaleźć w pracy [79, twierdzenie 5.7], a odpowiednik dla szeregów Fouriera w podręczniku [32].

Oznaczmy przez $R_n[f]$ wielomian optymalny najlepiej aproksymujący w normie jednostajnej funkcję f na przedziale $[-1, 1]$ (wielomian ten można znaleźć, stosując algorytm Remez [17, podrozdz. 3.8]). Można dowieść, że błąd aproksymacji sumami Czebyszewa $S_n[f]$ w ogólności nie może różnić się od aproksymacji wielomianem $R_n[f]$ bardziej niż o czynnik $\frac{4}{\pi^2} \log(n) + 1.2703 + \mathcal{O}(1/n)$ (co dla $n = 100\,000$ jest mniejsze niż 6) [79, rozdz. 7].

W przypadku zbieżności w normie $\|\cdot\|_1$ dla funkcji ciągłej określonej na przedziale zwartym powyższe twierdzenia pozostają prawdziwe z tą różnicą, że ograniczenia są mnożone przez stały czynnik równy $\|f\|_1$, który dla funkcji gęstości wynosi 1 [79]. Rozszerzenie tych rezultatów na przypadek funkcji określonej na przedziałach nieskończonych oraz na funkcje z osobliwościami nie jest bezpośrednie. W szczególności interesująca z punktu widzenia obliczeń probabilistycznych aproksymacja dowolnej funkcji gęstości na całej prostej rzeczywistej za pomocą wielomianów nie jest możliwa.

4.5.2. Dokładność obliczeń zmiennopozycyjnych

Teoretyczne ograniczenia na jakość aproksymacji trzeba dodatkowo łączyć z ograniczeniami obliczeń zmiennopozycyjnych. Obliczenia numeryczne z użyciem liczb w zapisie zmiennopozycyjnym są oczywiście obciążone błędami związanymi ze skończoną reprezentacją. Obecnie standardowo używa się zmiennych podwójnej precyzji (`double` zgodnie ze standardem IEEE 754 [48]), które są zapisywane na 64 bitach, z czego mantysa składa się z 52 bitów, a cecha z 11 bitów oraz 1 bitu znaku. Ponadto norma IEEE 754 (podobnie jak większość kompilatorów) określa sposoby posługiwania się symbolami: `+0`, `-0`, `-Inf`, `+Inf`, `NaN`. Pojawienie się w trakcie obliczeń któregoś z powyższych symboli zwykle wymaga specjalnego fragmentu kodu do obsługi takiej sytuacji.

Dokładność obliczeń zmiennopozycyjnych zależy od precyzji argumentów. Precyzję maszynową ϵ (precyzję obliczeń zmiennopozycyjnych) określa się jako ograniczenie górne na błąd względny reprezentacji dowolnej liczby w zapisie zmiennopozycyjnym (oznaczonej $\hat{x}$):

$$\frac{|\hat{x} - x|}{|x|} \leq \epsilon.$$

W przypadku liczb typu `double` wartość $\epsilon = \frac{1}{2} \cdot 2^{52} \approx 1.11 \times 10^{-16}$, co odpowiada praktycznie 15 lub 16 znaczącym cyfrom dziesiętnym. Definiuje się również funkcję `eps(x)`, która zwraca odległość do kolejnej najbliższej liczby reprezentowanej w typie `double`. Wielkość ta jest z kolei ograniczeniem na błąd bezwzględny [48, 64, 116]. Dokładność obliczeń zmiennopozycyjnych jest mierzona w terminach precyzji maszynowej, która jest granicą

dokładności obliczeń zmiennopozycyjnych, przy czym w praktyce ważne jest utrzymanie zarówno dokładności względnej, jak i bezwzględnej. Utrzymanie wysokiej dokładności względnej jest w przypadku bardziej złożonych obliczeń zwykle trudne do wykonania.

Dokładność sumy/różnicy dwóch liczb zmiennopozycyjnych jest na poziomie dokładności bezwzględnej większego z argumentów; jeżeli natomiast argumenty różnią się o rzędy wielkości, to błąd względny może być bardzo duży (zwłaszcza przy odejmowaniu liczb o podobnym rzędzie wielkości). Wynika stąd, że dla sum nie powinniśmy oczekiwać dokładności bezwzględnej większej niż wartość funkcji eps dla większego z argumentów. W przypadku mnożenia i dzielenia dokładność obliczeń może być wyższa, gdyż działania te utrzymują dokładność względną. Szczegółowe omówienie zagadnień związanych z kontrolą dokładności obliczeń zmiennopozycyjnych można znaleźć w pracach [64, 116]. W przypadku obliczania wartości standardowych funkcji, jak: log, exp, sin czy cos, dokładność względna jest na poziomie kilkukrotnie większym niż wartość ϵ . Poniższy fragment kodu w programie Matlab szacuje dokładność względną obliczeń dla funkcji wykładniczej i logarytmu.

```
1 t=logspace(-100,100,1000000);
2 err_rel=max((t-exp(log(t)))./t)
```

```
err_rel =
```

```
1.434026478602391e-014
```

Na tej podstawie oczekiwanie dokładności na poziomie 1×10^{-14} przy wykonywaniu bardziej złożonych obliczeń wydaje się rozsądnym wymaganiem. Dokładność obliczeń na takim poziomie będzie dalej nazywana pełną dokładnością numeryczną.

Do oceny jakości aproksymacji są wykorzystywane różnice bezwzględne pomiędzy funkcją oryginalną f a jej aproksymacją $S_n[f]$:

$$err_{ABS}(x) = |f(x) - S_N[f](x)|$$

oraz analogiczne różnice względne:

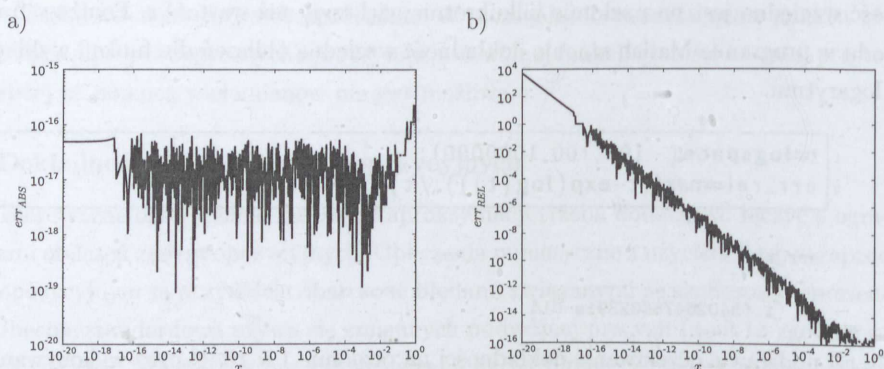
$$err_{REL}(x) = \left| \frac{f(x) - S_N[f](x)}{f(x)} \right|.$$

W dalszej części $err_{ABS} = \max_x err_{ABS}(x)$ oraz $err_{REL} = \max_x err_{REL}(x)$ oznaczają błąd względny i błąd bezwzględny. Dla poprawnie aproksymowanej funkcji oba rodzaje błędów powinny być na tym samym poziomie w całym zakresie dziedziny funkcji f . W przypadku błędów względnych maksymalna możliwa do uzyskania jakość aproksymacji jest na poziomie precyzji maszynowej ϵ , natomiast dokładność bezwzględna na poziomie $\epsilon \|f\|_\infty$.

4.5.3. Ograniczenia formuły barycentrycznej

Ograniczenia arytmetyki zmiennopozycyjnej rzutują na dokładność obliczenia formuły barycentrycznej, która ma postać ilorazu sum. Rozważmy przykład z obliczaniem wartości funkcji sinus.

Przykład 4.1. Niech $f(x) = \sin(x)$ dla $x \in [0, 1]$. Zbudowano interpolator na węzłach Czebyszewa drugiego rodzaju, aproksymujący funkcję f z dokładnością bezwzględną mniejszą niż 16×10^{-16} . Można sprawdzić, że w tym przypadku wystarczy 17 węzłów. Następnie porównano błędy numeryczne interpolatora. Na rysunku 4.7a, b przedstawiono porównanie błędu bezwzględnego i błędu względnego dla funkcji sinus. Widać, że błąd bezwzględny, zgodnie z oczekiwaniami, jest na poziomie 1×10^{-15} , natomiast błąd względny rośnie dla argumentów bliskich zera.



Rys. 4.7. Błąd interpolacji barycentrycznej w skali logarytmicznej dla funkcji sinus w przedziale $[0, 1]$, z użyciem 17 węzłów: a) błąd bezwzględny (err_{ABS}), b) błąd względny (err_{REL})

Przykład ten pokazuje, że utrzymanie wysokiej dokładności względnej dla funkcji zmieniających się o wiele rzędów wielkości z wykorzystaniem formuły barycentrycznej może powodować pewne trudności. Aby aproksymować funkcję z zachowaniem wysokiej dokładności względnej w otoczeniu zera, należy mieć dodatkową informację na temat asymptotycznego zachowania się funkcji w otoczeniu zera. Należy dodać, że dla funkcji cosinus na przedziale $[0, \frac{\pi}{2}]$ nie zaobserwujemy aż tak dużej utraty dokładności w otoczeniu miejsca zerowego $x = \frac{\pi}{2}$. Jest to spowodowane tym, że w otoczeniu punktu $\frac{\pi}{2}$ liczby zmiennopozycyjne są rozłożone znacznie rzadziej niż w otoczeniu zera¹⁰.

Warto zaznaczyć, że chociaż w przykładzie 4.1 interpolator nie zachowuje dokładności względnej w otoczeniu zera, nie ma to jednak wpływu na dokładność obliczenia całki. W praktyce utrzymywanie wysokiej dokładności względnej jest trudniejsze do osiągnięcia

¹⁰ $\text{eps}(0) = 4.9406 \times 10^{-324}$, $\text{eps}(1) = 2.2204 \times 10^{-16}$.

i zwykle nie jest konieczne, na przykład w obszarach, gdzie funkcja przyjmuje wartości bliskie zera. W większości przypadków, zwłaszcza dla funkcji $f \in C[a, b]$, wystarcza wysoka dokładność bezwzględna. Utrzymywanie wysokiej dokładności względnej jest ważne dla funkcji, które zmieniają się o wiele rzędów wielkości, na przykład w otoczeniu punktów osobliwych lub w nieskończoności.

4.6. Aproksymacja funkcji gęstości, funkcje kawałkami gładkie

4.6.1. Funkcje na przedziale skończonym

Ponieważ nie istnieją ogólne metody umożliwiające aproksymację dowolnych funkcji ciągłych, w praktyce należy się ograniczyć do pewnych mniej lub bardziej szczególnych przypadków. W dalszej części ograniczamy się do aproksymacji funkcji dodatnio określonych oraz całkowalnych (funkcji gęstości), które składają się ze skończonej liczby gładkich segmentów. Wybór ten jest motywowany twierdzeniem 2.1.

Punkty graniczne segmentów (punkty nieciągłości funkcji lub jej pochodnych) będziemy nazywać dalej *punktami potencjalnie osobliwymi*. Jeżeli w otoczeniu pewnego punktu nie da się poprawić własności analitycznych funkcji przez rozbiecie dziedziny na podsegmenty, to taki punkt będzie zwany *punktem osobliwym*.

Różnicę pomiędzy punktem osobliwym a punktem potencjalnie osobliwym widać na przykładzie dwóch funkcji: $f(x) = |x|$ oraz $g(x) = 1/x$, określonych, odpowiednio, na przedziałach $[-1, 1]$ oraz $[-1, 0) \cup (0, 1]$. Funkcja f jest klasy $C^0[-1, 1]$ i jest to ogólnie funkcja trudna do aproksymacji. Jeżeli jednak dziedzinę tej funkcji rozbijemy na dwa przedziały: $[-1, 0]$ oraz $[0, 1]$, to w każdym z tych przedziałów f będzie funkcją analityczną. Na podstawie twierdzenia 4.1 funkcje takie są znacznie prostsze do aproksymacji z użyciem wielomianów Czebyszewa. W tym przypadku punkt $x = 0$ jest punktem potencjalnie osobliwym funkcji f . Natomiast rozbiecie dziedziny funkcji g w otoczeniu zera nie poprawi własności analitycznych tej funkcji w otoczeniu zera, punkt ten pozostanie osobliwy.

Kawałkami gładką funkcję f można przedstawić w postaci skończonej sumy segmentów:

$$f(x) = \sum_{i=1}^n \chi_{[a_i, b_i]} f_i(x), \quad f_i \in C^k[a_i, b_i] \quad \text{dla } i = 1, \dots, n.$$

Z technicznego punktu widzenia należy rozważyć następujące typy segmentów:

- funkcja klasy $C^k[a, b]$,
- funkcja klasy $C^k(a, b]$ mająca osobliwość lewostronną w punkcie a ,
- funkcja klasy $C^k[a, b)$ mająca osobliwość prawostronną b ,
- funkcja klasy $C^k(-\infty, b]$ na przedziale lewostronnie nieskończonym,
- funkcja klasy $C^k[a, \infty)$ na przedziale prawostronnie nieskończonym.

Bezpośrednio używając wielomianów Czebyszewa, można aproksymować funkcję na przedziale $[-1, 1]$. Jeżeli funkcja jest określona ogólnie na przedziale $[a, b]$, należy zastosować odpowiednią transformację zmiennych, aby sprowadzić rozważania do przedziału $[-1, 1]$. Dotyczy to zarówno przedziałów skończonych, jak i nieskończonych. Skończony przedział $[a, b]$ możemy transformować do przedziału $[-1, 1]$ na wiele sposobów. Jeżeli a i b są tego samego rzędu (to znaczy $\frac{b}{a} \approx 1$), to najwłaściwsza wydaje się transformacja liniowa:

$$v_{\text{lin}}(x) = \frac{x - \frac{a+b}{2}}{b - a}.$$

Z drugiej strony, jeżeli $\frac{b}{a} \gg 1$ (lub $\frac{a}{b} \gg 1$)¹¹, użycie transformacji logarytmicznej:

$$v_{\text{log}}(x) = \frac{2 \log(x/\sqrt{ab})}{\log(b/a)}$$

wyda się przekształceniem lepiej dostosowanym do obliczeń zmiennopozycyjnych¹².

Zastosowanie transformacji liniowej i formuły barycentrycznej jest bezpośrednie, wymaga jedynie przeskalowania węzłów Czebyszewa do przedziału $[a, b]$. Dalsze obliczenia przebiegają analogicznie jak w przypadku zwykłej interpolacji barycentrycznej. Transformacje nieliniowe wymagają zastosowania nieco bardziej złożonej procedury. W takim przypadku interpoluje się odpowiednio zmodyfikowaną funkcję, natomiast do odtworzenia funkcji oryginalnej konieczne jest zastosowanie transformacji odwrotnej. Transformacje dla przedziałów otwartych zostaną omówione w dalszej części.

4.6.2. Funkcje na przedziałach nieskończonych, funkcje z osobliwościami

W przypadku przedziałów nieskończonych można stosować wiele różnych metod zamiany zmiennych, podobnie jak przy liczeniu całek na przedziałach nieskończonych [12, 14, 22]. W pakiecie **PaCAL** są wykorzystywane przekształcenia oparte na funkcjach wymiernych w postaci:

$$v(x) = \left(\frac{A}{x - a + A} \right)^{1/\gamma}, \quad v^{-1}(t) = a - A + \frac{A}{t^\gamma}. \quad (4.23)$$

Parametry tych funkcji zależą od punktów brzegowych dziedziny i są dobierane w ten sposób, aby przekształcić zadany przedział półnieskończony (na przykład $[a, \infty)$) w przedział $[0, 1]$ ¹³. Wykładnik $\gamma \in [1, \infty)$ może być w większości przypadków ustawiony na 1. Natomiast większe wartości wykładnika mają duże znaczenie dla funkcji mających ciężkie

¹¹ Zakładamy, że a i b są jednego znaku ($a < b < 0$ lub $0 < a < b$).

¹² Liczby zmiennopozycyjne są równomiernie rozłożone w skali logarytmicznej.

¹³ Domknięcie w zerze wynika z faktu, że $v(\infty) = 0$.

ogony (podrozdz. 4.4.2 oraz przykład 4.8). W bibliotece **PaCAL** wartość tego wykładnika γ jest ustawiona na 6, co teoretycznie zapewnia możliwość aproksymacji funkcji zachowujących się asymptotycznie jak $\mathcal{O}(\frac{1}{x^{1+1/6}})$. W praktyce obejmuje to nawet takie rozkłady, jak rozkład Levy'ego. Praca z jeszcze cięższymi ogonami jest możliwa po zwiększeniu wykładnika (przykład 4.8). Procedura interpolacji przebiega w ten sposób, że interpolowana jest funkcja $g(t) = f(v^{-1}(t))$ dla $t \in [-1, 1]$, w wyniku otrzymuje się interpolator $S_N[g]$. Natomiast wartość funkcji interpolowanej jest obliczana z użyciem ponownej transformacji zmiennej $S_n[g](v(x))$ dla $x \in [a, \infty)$.

Jedna ze stosowanych technik aproksymacji funkcji z osobliwościami polega na założeniu, że aproksymowana funkcja daje się przedstawić w postaci¹⁴:

$$f(t) = x^\alpha g(x), \quad x \in (0, b], \quad (4.24)$$

gdzie $g(x)$ jest funkcją ciągłą na przedziale domkniętym $[0, b]$. W takim przypadku pozostają do wykonania dwie rzeczy: aproksymacja funkcji $g(x)$ oraz oszacowanie wykładnika α . W trakcie prac nad projektem **PaCAL** wykonano szereg testów z użyciem funkcji tego typu, dochodząc do wniosku, że w wielu przypadkach takie podejście nie jest najlepsze. Podstawowy problem dotyczy oszacowania wykładnika α . Jeżeli wykładnik ten nie zostanie oszacowany dokładnie, to aproksymowana funkcja będzie miała postać: $g(x) = x^{\alpha - \hat{\alpha}} g(x)$. Nietrudno sprawdzić, że dla $\alpha \neq \hat{\alpha}$ funkcja ta ma pochodną nieograniczoną w otoczeniu zera. Jeżeli wykładnik nie jest zidentyfikowany bardzo dokładnie, to przy interpolacji takich funkcji mogą pojawić się dość znaczne błędy. Drugim mankamentem faktoryzacji postaci (4.24) jest fakt, że istnieją osobliwości innego typu niż potęgowe, jak na przykład funkcja logarytmiczna – mająca w zerze osobliwość innego typu.

Estymację wykładnika α w punkcie x można wykonywać, stosując równość

$$\frac{f(t_1) - f(t_2)}{(t_1 - t_2)^\alpha} \approx 1, \quad \text{dla } |t_1 - x| + |t_2 - x| < \epsilon,$$

którą po zlogarytmowaniu można zapisać w postaci:

$$\alpha = \frac{\log(f(t_1) - f(t_2))}{\log(t_1 - t_2)}.$$

Podejście to działa dość dobrze, niemniej istnieje pewien problem z wyborem odpowiednich granicznych wartości t_1 i t_2 . Podobnie można szacować wykładnik w nieskończoności.

¹⁴ Podejście takie jest na przykład stosowane w bibliotece **chebfun**, ponadto biblioteka ta umożliwia adaptacyjne rozcinanie przedziałów w otoczeniu punktu osobliwego.

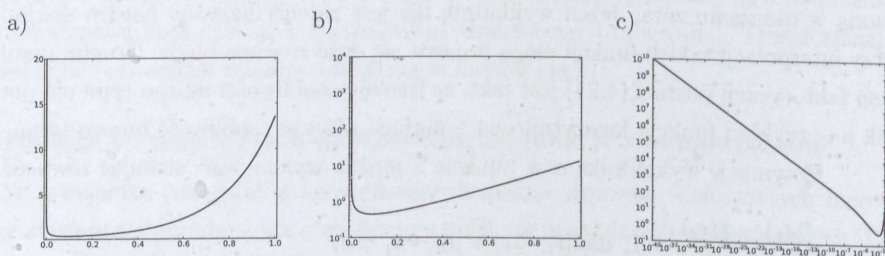
Transformacje logarytmiczno-logarytmiczne

Jednym ze sposobów zmiany skali wartości funkcji zmieniających się o wiele rzędów wielkości jest zastosowanie transformacji logarytmicznej. Z zależności:

$$\log(x \cdot (1 + \epsilon)) = \log(x) + \log(1 + \epsilon) \approx \log(x) + \epsilon$$

wynika, że logarytmowanie może zamienić dokładność bezwzględną w dokładność względną. Ma to znaczenie zarówno przy aproksymacji funkcji w otoczeniu punktów osobliwych, jak i w przypadku, gdy ważne jest zachowanie wysokiej dokładności względnej (na przykład w otoczeniu zera). Należy jednak mieć świadomość, że samo logarytmowanie nie zawsze prowadzi do prostszej funkcji (przy aproksymacji), co widać na przykładzie logarytmowania wielomianów. Zasadniczą motywacją do stosowania różnego rodzaju transformacji zmiennych jest chęć uzyskania prostszej funkcji do aproksymacji. Pewną ilustrację stanowi tu następujący przykład:

Przykład 4.2. Rozważmy funkcję $h(x) = \frac{\exp(5x)}{10x^{0.5} + x^{0.25}}$ dla $x \in (0, 1]$. Ma ona punkt osobliwy ($x = 0$). Na rysunku 4.8 przedstawiono wykresy tej funkcji w różnych skalach: liniowej, logarytmicznej na osi rzędnych oraz logarytmicznej na obu osiach. Widać, że najbardziej regularna postać funkcji (najmniej stromy wykres) jest na skali podwójnie logarytmicznej, w tej skali zmiany funkcji są najmniejsze.



Rys. 4.8. Wykresy funkcji $h(x) = \frac{\exp(5x)}{10x^{0.5} + x^{0.25}}$, $x \in (0, 1]$ w różnych skalach: a) liniowej, b) mieszanej, c) logarytmicznej

W pakiecie **PaCAL** w otoczeniu punktów osobliwych są stosowane transformacje podwójnie logarytmiczne, wykorzystujące zarówno logarytmowanie dziedziny funkcji, jak i zmianę skali wartości funkcji. Transformacje mają postać:

$$\begin{aligned} v_X(x) &= \log(x), & v_X^{-1}(x) &= \exp(x), \\ v_Y(x) &= \log1p(x) = \log(x + 1), & v_Y^{-1}(x) &= \expm1(x) = \exp(x) - 1. \end{aligned}$$

Wykorzystano funkcje `log1p` i `expm1`, aby zachować wysoką dokładność względną w oto-

czeniu zera; szczegóły implementacyjne tych funkcji można znaleźć w pracy [116, podrozdz. D]. Funkcje te odwzorowują bijektywnie przedział $[0, \infty)$ na samego siebie.

Ostatecznie dla danej funkcji f określonej na przedziale $(a, b]$ oraz mającej osobliwość w punkcie a interpolowana funkcja ma postać:

$$g(z) = v_Y\left(f(v_X^{-1}(z))\right) \quad \text{dla } x \in [v_X(a + \epsilon), v_X(b)],$$

gdzie ϵ jest małą liczbą, rzędu $\text{eps}(a)$. Dla $a = 0$ przyjęto wartość $\epsilon = 1 \times 10^{-50}$, co gwarantuje wysoką dokładność w większości testowanych przypadków. Mając interpolator $S_N[g]$, wartość funkcji $f(x)$ aproksymujemy, używając transformacji odwrotnej:

$$f(x) \approx v_Y^{-1}\left(S_N[g](v_X(x))\right) \quad \text{dla } x \in [a, b].$$

Podejście takie wybrano, testując empirycznie metody różnego rodzaju. Niektóre przykłady pokazano w dalszej części (przykład 4.7).

4.6.3. Aproksymacja w klasie funkcji $\mathcal{F}$

Jeżeli ograniczymy się do funkcji z klasy $\mathcal{F}_k$ wprowadzonej w podrozdz. 2.8, to funkcje takie można aproksymować z użyciem wielomianów oraz odpowiednich transformacji zmiennych.

Twierdzenie 4.2. Dla każdej funkcji f z rodziny $\mathcal{F}_k$ istnieje funkcja kawałkami wielomianowa (ewentualnie z zastosowaniem transformacji zmiennych) przybliżająca jednostajnie f z dowolną dokładnością na całej dziedzinie z pominięciem dowolnie małych otoczeń punktów osobliwych. Dla $k \geq 1$ funkcję aproksymującą można uzyskać za pomocą interpolacji Czebyszewa. Jeśli dodatkowo $k \geq 2$, można podać górne ograniczenie na tempo zbieżności aproksymacji.

Twierdzenie zostało udowodnione w pracy [51]; przy dowodzie wykorzystuje się własności potęgowych transformacji zmiennych oraz twierdzenie 4.1.

4.7. Pakiet **chebfun**

Wiele operacji numerycznych na dowolnych funkcjach ciągłych można w praktyce wykonywać, posługując się bezpośrednio skończonymi reprezentacjami tych funkcji. Obliczenia takie mogą być wykonywane w analogiczny sposób jak zwykle obliczenia na wektorach i macierzach. Praktyka pokazuje, że takie operacje mogą być wielokrotnie składowane z zachowaniem wysokiej dokładności obliczeń. Koncepcję tę wykorzystano w pakiecie **chebfun** [5, 120] programu Matlab. Ogólna idea obliczeń w pakiecie **chebfun** jest taka, że typowe dla analizy funkcjonalnej działania na funkcjach ciągłych wykonuje się w sposób

analogiczny jak w skończenie wymiarowej algebrze liniowej (algebrze macierzy i wektorów). Projekt ten wykorzystuje w większości znane wcześniej metody numeryczne, jednak zasadnicza wartość projektu polega na wprowadzeniu jednolitej reprezentacji numerycznej funkcji ciągłych oraz operatorów [5, 10]. Dzięki takiemu podejściu wyniki częściowe kolejnych operacji mogą być ze sobą składane oraz brać udział w kolejnych operacjach. Biblioteka obsługuje przypadek funkcji jednej zmiennej, obejmujący również zmienną zespoloną. W nowszych wersjach są również wspierane obliczenia na funkcjach określonych na przedziale nieskończonym oraz funkcje z osobliwościami. Pakiet ten umożliwia wykonywanie obliczeń w sposób funkcjonalnie zbliżony do obliczeń symbolicznych, jednak opierając się na metodach czysto numerycznych.

Z punktu widzenia obliczeń probabilistycznych pakiet **chebfun** ma w zasadzie tylko jedną operację „probabilistyczną” – splot (funkcja `conv`) oraz ewentualnie całkę skumulowaną (`cumsum`). Porównania i testy numeryczne pokazują (podrozdz. 6.2.3), że operacja splotu jest znacznie lepiej wykonywana w pakiecie **PaCAL**. Pakiety różnią się znacznie pod względem reprezentacji funkcji na przedziałach nieskończonych i funkcji z osobliwościami. Aktualna wersja biblioteki **chebfun** (z 2012 r.) nie umożliwia liczenia splotów dla funkcji określonych na przedziałach nieskończonych. W bibliotece **chebfun** w otoczeniu punktów osobliwych jest wykorzystywany adaptacyjny podział na większą liczbę przedziałów w otoczeniu punktu osobliwego do momentu uzyskania odpowiedniej jakości aproksymacji, dodatkowo są szacowane wykładniki w otoczeniu punktów osobliwych. Takie rozwiązania były również analizowane w trakcie prac nad projektem **PaCAL**, jednak znacznie lepsze wyniki dawały opracowane transformacje zmiennych. Podobnie wygląda sytuacja dla przedziałów nieskończonych. Procedury pakietu **PaCAL** są w stanie utrzymywać dokładność nawet dla bardzo ciężkich ogonów (por. przykłady 4.7 i 4.8), a dokładność aproksymacji jest na tyle duża, że możliwe jest nie tylko zapewnienie dokładności bezwzględnej (zbieżność jednostajna), ale i względnej, co pozwala na przykład na dokładne oszacowanie wykładnika w nieskończoności (por. przykład dotyczący rozkładów ilorazowych – podrozdz. 6.2.4, s. 132).

Należy podkreślić jednak, że główne cele obu bibliotek są odmienne. Pakiet **chebfun** jest pomyślany jako ogólne narzędzie do obliczeń numerycznych, jak całkowanie i różniczkowanie, ale również rozwiązywanie równań różniczkowych i całkowych. Natomiast pakiet **PaCAL** jest specjalnie zaprojektowany pod kątem obliczeń probabilistycznych. Niemniej biblioteka ta była jednym z ważnych punktów odniesienia podczas testowania reprezentacji numerycznych. Jest to w zasadzie jedyna ogólna biblioteka oparta na numerycznych reprezentacjach funkcji ciągłych (historycznie pierwsza), pozwalająca na pewne porównania. Biblioteki typu **polynomial.chebyshev** (dostępna w pakiecie **numpy** języka Python, podobne biblioteki można spotkać w innych językach) mają nieco ograniczone możliwości: nie umożliwiają liczenia splotów, również nie obsługują przedziałów

nieskończonych czy funkcji z osobliwościami. Oczywiście w zakresie całkowania i interpolacji funkcji ciągłych na przedziałach domkniętych są to rozwiązania równoważne.

4.8. Obliczenia numeryczne w pakiecie PaCAL

Metody numeryczne na funkcjach ciągłych, opisane w tym podrozdziale, zostały praktycznie zaimplementowane w pakiecie **PaCAL** w modułach: `pacal.interpolation`, `pacal.integration` oraz `pacal.segments`. Dostępne klasy interpolatorów zostały zdefiniowane w module `pacal.interpolation` (tab. 4.2).

Tab. 4.2. Interpolatory zaimplementowane w module `pacal.interpolation`

Metoda	Opis
<code>Interpolator(Xs,Ys)</code>	definiuje wielomian interpolacyjny Lagrange'a na węzłach X_s, Y_s
<code>BarycentricInterpolator(Xs,Ys,ws)</code>	interpolacja barycentryczna z ustalonymi wagami w_s , na węzłach X_s, Y_s , w ogólności prowadzi do interpolacji wymiernej
<code>ChebyshevInterpolator(f,a,b)</code>	interpolacja barycentryczna na węzłach Czebyszewa drugiego rodzaju funkcji f , na przedziale $[a, b]$
<code>ChebyshevInterpolator1(f,a,b)</code>	interpolacja barycentryczna na węzłach Czebyszewa pierwszego rodzaju (bez punktów brzegowych) funkcji f , na przedziale $[a, b]$, wygodna dla funkcji nieokreślonych na końcach przedziału
<code>PoleInterpolatorP(f,a,b)</code>	interpolator z transformacją logarytmiczną lewostronną dla funkcji z osobliwością na lewym końcu przedziału
<code>PoleInterpolatorN(f,a,b)</code>	interpolator z transformacją logarytmiczną lewostronną dla funkcji z osobliwością na prawym końcu przedziału
<code>ChebyshevInterpolator_PInf(f,L)</code>	interpolator z transformacją zmiennych dla funkcji na przedziale $[L, \infty)$
<code>ChebyshevInterpolator_MInf(f,U)</code>	interpolator dla funkcji na przedziale półnieskończonym $(-\infty, U]$

Klasy te wykorzystują interpolację barycentryczną z różnymi zestawami węzłów Czebyszewa oraz transformacje zmiennych w przypadku przedziałów nieskończonych i funk-

cji z osobliwościami. Klasy te odpowiadają wszystkim rodzajom segmentów wymienionych w podrozdz. 4.6.

W module `pacal.integration` zdefiniowano metody do całkowania przedstawione w tab. 4.3.

Tab. 4.3. Metody całkowania zaimplementowane w pakiecie `pacal.integration`

Metoda	Opis
<code>integrate.clenshaw(f,a,b)</code>	metoda Clenshawa–Curtisa
<code>integrate.fejer2(f,a,b)</code>	metoda Fejéra drugiego rodzaju (analogicznie jak Clenshawa–Curtisa tylko bez punktów brzegowych) – podstawowy integrator biblioteki PaCAL
<code>integrate.fejer2.Xn.transformP(f,a,b)</code>	metoda całkowania z zamianą zmiennych, dostosowana do funkcji z osobliwością w punkcie a
<code>integrate.fejer2.Xn.transformN(f,a,b)</code>	metoda całkowania z zamianą zmiennych, dostosowana do funkcji z osobliwością w punkcie b
<code>integrate.fejer2.pinf(f,a)</code>	metoda całkowania z zamianą zmiennych, dostosowana do przedziałów prawostronnie nieskończonych
<code>integrate.fejer2.minf(f,b)</code>	metoda całkowania z zamianą zmiennych, dostosowana do przedziałów lewostronnie nieskończonych

Klasy modułu `pacal.segments` dostarczają reprezentacji numerycznych dla funkcji kawałkami gładkich; wykorzystywane są one następnie przez klasy i metody probabilistyczne opisane w rozdziale 5. Główne klasy modułu zebrano w tab. 4.4, natomiast ważniejsze metody podstawowej klasy `PiecewiseFunction` przedstawiono w tab. 4.5.

W dalszej części zostaną przedstawione przykłady numeryczne ilustrujące efektywność opisanych metod numerycznych wykorzystujących wielomiany Czebyszewa oraz transformacje zmiennych. Pokazane zostaną również porównania z innymi rozwiązaniami numerycznymi.

Tab. 4.4. Klasy reprezentujące funkcje kawałkami gładkie (moduł `pacal.segments`)

Klasa	Opis
<code>PiecewiseFunction</code>	funkcja kawałkami gładka, parametry konstruktora to: <code>fun</code> – dana funkcja; <code>breakPoints</code> – lista punktów podziału dziedziny (punktów nieciągłości lub punktów osobliwych) włączając w to symbole <code>-Inf</code> i <code>Inf</code> ; <code>lpoles</code> – lista o długości takiej samej jak lista <code>breakPoints</code> zawierająca wartości <code>True</code> lub <code>False</code> oznaczająca, że w danym punkcie jest osobliwość lewostronna; <code>rpoles</code> – analogiczna lista do oznaczenia osobliwości prawostronnych
<code>PiecewiseDistribution</code>	podklasa reprezentująca funkcje gęstości; w klasie tej zdefiniowano dodatkowe metody poprawnie określone jedynie dla funkcji gęstości

Tab. 4.5. Ważniejsze metody zdefiniowane w klasie `PiecewiseFunction`

Metoda	Opis
<code>integrate(a,b)</code>	całka oznaczona na przedziale $[a, b]$; końce przedziału mogą być niewłaściwe
<code>cumint()</code>	całka skumulowana reprezentowana przez obiekt <code>PiecewiseFunction</code> – funkcja określona wzorem $F(x) = \int_{-\infty}^x f(\tau) d\tau$
<code>ccumint()</code>	komplementarna całka skumulowana – funkcja określona wzorem $F(x) = \int_x^{\infty} f(\tau) d\tau$
<code>diff()</code>	pochodna funkcji
<code>roots()</code>	lista miejsc zerowych funkcji
<code>toInterpolated()</code>	interpolacja funkcji
<code>trimInterpolators(abstol)</code>	obniżanie rzędu interpolatora do zadanej tolerancji <code>abstol</code> z wykorzystaniem rozwinięcia w szereg Czebyszewa
<code>min_()</code>	minimum funkcji
<code>max_()</code>	maksimum funkcji
<code>max_abs()</code>	maksimum z wartości bezwzględnej
<code>plot(xmin, xmax)</code>	wykres funkcji

4.9. Przykłady numeryczne

4.9.1. Funkcje ciągłe na przedziale domkniętym

Twierdzenia o aproksymacji wielomianami Czebyszewa pokazują, że w przypadku funkcji analitycznych uzyskujemy wykładnicze tempo zbieżności aproksymacji, natomiast dla funkcji k -gładkich (mających k ciągłych pochodnych) tempo zbieżności jest wielomianowe, przy czym stopień wielomianu zależy od stopnia liczby ciągłych pochodnych. Poniższe dwa przykłady pokazują aproksymację funkcji tego typu.

Przykład 4.3 (interpolacja funkcji analitycznych). Rozważmy trzy funkcje:

- $f_1(x) = \sin(1/x), x \in [0.01, 1]$,
- $f_2(x) = \sin(1/x), x \in [0.1, 10]$,
- $f_3(x) = \log(x), x \in [0.1, 10]$.

Dla każdej z powyższych funkcji utworzono interpolator barycentryczny na węzłach Czebyszewa zgodnie z procedurą adaptacyjnego podwajania węzłów (alg. 1). Interpolatory budowano do momentu, aż maksymalny błąd bezwzględny osiągnął wartość mniejszą niż 16×10^{-16} . Następnie za pomocą szybkiej transformaty Fouriera (podrozdz. 4.3.1) wyznaczono współczynniki wielomianu aproksymacyjnego zapisanego w bazie Czebyszewa. Dokonując odcięcia wszystkich współczynników poniżej pewnej wartości ϵ , ustalono stopień wielomianu aproksymacyjnego. Dla ustalonego stopnia wielomianu ponownie wyznaczono dokładny błąd aproksymacji, próbując odpowiednio gęsto funkcje w zadanym przedziale. Każda z funkcji ma pochodną dowolnego rzędu. Jak widać na rys. 4.9, logarytm z błędu w normie jednostajnej maleje od pewnego momentu liniowo wraz ze wzrostem liczby węzłów wielomianu interpolacyjnego. Punkty na wykresach odpowiadają wartościom $\epsilon = 10^{-1}, 10^{-2}, \dots, 10^{-14}$. Dokładniejsza analiza pokazuje, że zmienność funkcji ma wpływ na liczbę węzłów interpolatora. Na przykład funkcja f_1 ma pochodną $f_1'(x) = -\frac{1}{x^2} \cos(1/x)$, która w punkcie $x = 0.01$ przyjmuje wartość rzędu 10^4 , mimo że sama funkcja f_1 zmienia się w zakresie $[-1, 1]$. Skutkuje to większą liczbą węzłów interpolacyjnych potrzebną do osiągnięcia zadanego poziomu aproksymacji.

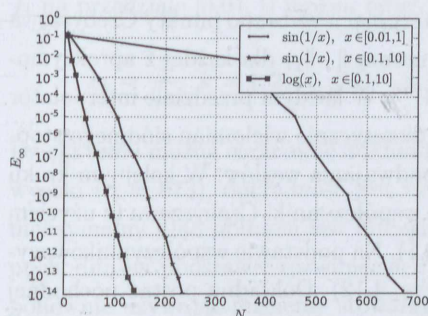
Kolejny przykład pokazuje podobny eksperyment przeprowadzony dla bardziej nieregularnych funkcji.

Przykład 4.4 (interpolacja funkcji klasy C^k). Rozważmy następujące funkcje:

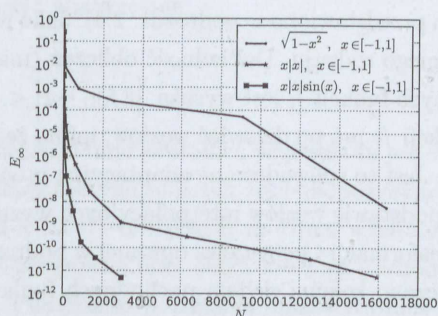
- $g_1(x) = \sqrt{1-x^2}, x \in [-1, 1]$,
- $g_2(x) = x|x|, x \in [-1, 10]$,
- $g_3(x) = x|x| \sin(x), x \in [0.1, 10]$.

Wszystkie trzy funkcje są ciągłe w swoich dziedzinach. Pierwsza z nich, g_1 , nie należy do C^1 . Funkcja ta ma pochodną ciągłą na przedziale $(-1, 1)$ (pochodna nie jest określona na końcach przedziału), jednak pochodna ta jest nieograniczona. Skutkuje to tym, że funkcja ta jest najtrudniejsza do aproksymacji. Przy liczbie węzłów interpolatora około

16 000 dokładność w normie jednostajnej jest rzędu 10^{-8} . Funkcja g_2 ma dobrze określoną pochodną ($g_2'(x) = |x|$), jednak druga pochodna tej funkcji nie jest określona dla $x = 0$. Trzecia funkcja, g_3 , jest klasy C^2 , a trzecia pochodna tej funkcji ponownie nie jest określona dla $x = 0$. Wyniki identycznego eksperymentu jak dla funkcji f_i przedstawiono na rys. 4.10. Jak widać na wykresie, im bardziej regularna funkcja, tym tempo zbieżności jest szybsze, jednak zbieżność w tym przypadku nie jest wykładnicza. Ponadto szacowanie dokładności na podstawie wartości odrzucanych współczynników jest w tym przypadku nieco mniej dokładne, odcięcie współczynników poniżej 10^{-14} daje faktyczną dokładność na poziomie 10^{-12} .



Rys. 4.9. Dokładność jednostajnej aproksymacji funkcji nieskończenie gładkich w zależności od liczby węzłów



Rys. 4.10. Dokładność jednostajnej aproksymacji funkcji klasy C^k w zależności od liczby węzłów

Splot jest bardziej złożoną numerycznie operacją niż całkowanie. Splot dwóch funkcji wymaga wielokrotnego całkowania. Dla dodatkowo określonych funkcji klasy L^1 splot jest operacją zachowującą normę, to znaczy:

$$\left\| \int_{-\infty}^{\infty} (f(t - \tau)g(\tau) d\tau) \right\|_1 = \|f\|_1 \|g\|_1.$$

Dzięki tej własności możemy kontrolować dokładność obliczeń, wykorzystując w tym celu normę L^1 funkcji wynikowej, równą dla funkcji gęstości 1. Kolejny przykład dotyczy obliczania splotów oraz różniczkowania.

Przykład 4.5 (splot i różniczkowanie). Rozważmy funkcję gęstości rozkładu jednostajnego na odcinku $[0, 2]$:

$$u(t) = \begin{cases} \frac{1}{2}, & t \in [0, 2], \\ 0, & t \notin [0, 2]. \end{cases}$$

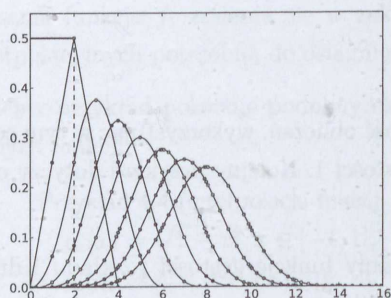
Zdefiniujmy ciąg funkcji określony następująco:

$$f_0 = u,$$

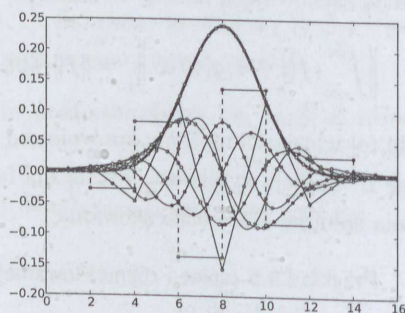
$$f_{i+1}(t) = \int_{-\infty}^{\infty} (f_i(t - \tau)u(\tau)) d\tau, \quad i = 1, 2, \dots$$

Nietrudno sprawdzić przez bezpośrednie całkowanie, że f_i jest funkcją kawałkami gładką z węzłami podziału w punktach $0, 2, \dots, 2(i+1)$ oraz że każdy segment jest wielomianem stopnia i . Ponadto z własności splotu wynika, że $\|f_i\|_1 = 1$. Na rysunku 4.11 przedstawiono wykresy funkcji dla $i = 0, 1, \dots, 7$, wraz z zaznaczonymi węzłami interpolacji. Całki spłotowe są liczone w sposób numeryczny (szczegóły liczenia splotów funkcji kawałkami gładkich przedstawiono w podrozdz. 2.4). Jako węzły interpolacji wybrano punkty Czebyszewa drugiego rodzaju. Dokładność obliczeń (mierzona normą $\|\cdot\|_1$) dla każdej z aproksymowanych funkcji f_i jest wysoka: $|\|f_i\|_1 - 1| < 16 \times 10^{-16}$. W każdym przedziale interpolator funkcji f_7 ma po dziewięć węzłów, mimo że jest to teoretycznie wielomian siódmego stopnia. Jest to spowodowane adaptacyjną procedurą podwajania węzłów. W kolejnym kroku na podstawie węzłów interpolacyjnych wyznaczono współczynniki Czebyszewa (z użyciem transformacji Czebyszewa opisanej w podrozdz. 4.3.1). Na podstawie współczynników wyznaczono kolejne siedem pochodnych funkcji f_7 (rys. 4.12). Dokładną postać pochodnej trzeba by wyznaczyć analitycznie, jednak bezpośrednio z wykresu można ocenić, że funkcje f_i są $i - 1$ -gładkie oraz wszystkie pochodne są ciągłe, z wyjątkiem pochodnej siódmego rzędu – pochodna szóstego rzędu jest kawałkami liniowa, a siódmego jest kawałkami stała. Jest to prawidłowy rezultat i wynika z własności splotu [35,44]. Dokładniej spłot poprawia własności analityczne funkcji w ten sposób, że $f_i \in C^{i-1}[0, 2(i+1)]$.

Kolejny przykład dotyczy obliczania pierwiastków i ekstremów funkcji.



Rys. 4.11. Siedmiokrotny spłot funkcji stałej na przedziale $[0, 2]$ oraz funkcje $f_0, f_1, \dots, f_7$ z przykładu 4.5



Rys. 4.12. Siedem kolejnych pochodnych funkcji f_7 z przykładu 4.5

Przykład 4.6 (zera i ekstrema funkcji $\sin(1/x)$). Rozważmy funkcję f_1 z przykładu 4.3:

$$f_1(x) = \sin(1/x), \quad x \in [0.01, 1].$$

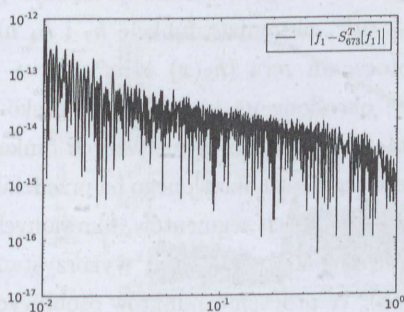
Procedura adaptacyjnego budowania interpolatora dla tej funkcji osiąga błąd poniżej 16×10^{-16} dopiero dla 1025 węzłów (po 10 iteracjach). Przejście do rozwinięcia w szereg Czebyszewa, a następnie usunięcie końcowych współczynników poniżej wartości 1×10^{-15} pozwala obniżyć stopień wielomianu do 673. Błąd aproksymacji przedstawiono na rys. 4.13. Na podstawie współczynników zbudowano macierz stowarzyszoną oraz wyznaczono wartości własne, które odpowiadają pierwiastkom funkcji f_1 (rys. 4.14). Pierwiastki funkcji f_1 na przedziale $[0.01, 1]$ można łatwo wyznaczyć analitycznie:

$$x_k = \frac{1}{k\pi}, \quad k = 1, 2, \dots, 31.$$

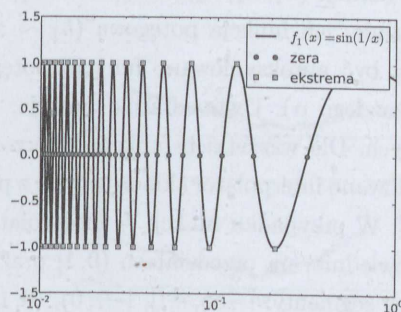
Dzięki temu można oszacować dokładność obliczeń. Mimo że stopień wielomianu jest dość wysoki ($n = 673$), dokładność jest równa pełnej precyzji obliczeń zmiennopozycyjnych, maksymalny błąd obliczeń jest mniejszy niż 4×10^{-15} . W następnym kroku wyznaczono pochodną na podstawie rozwinięcia w bazie Czebyszewa oraz obliczono zera pochodnej, ponownie używając macierzy stowarzyszonej. Pierwiastki pochodnej można również wyznaczyć analitycznie:

$$x_k = \frac{1}{(k + \frac{1}{2})\pi}, \quad k = 0, 1, \dots, 31.$$

Porównanie rozwiązań numerycznych z teoretycznymi pokazuje, że ponownie dokładność obliczeń jest bardzo wysoka – wszystkie pierwiastki pochodnej z wyjątkiem największego są wyznaczone z błędem poniżej 1×10^{-14} , natomiast największy jest pierwiastek obli-



Rys. 4.13. Różnica bezwzględna pomiędzy funkcją $f_1(x) = \sin(1/x)$ a funkcją interpolowaną $S_{673}[f_1]$



Rys. 4.14. Interpolacja funkcji $f_1(x) = \sin(1/x)$, $x \in [0.01, 1]$; zaznaczono obliczoną zera i ekstrema funkcji

czony z dokładnością 2.22×10^{-12} . Utrata dokładności w tym przypadku jest spowodowana najprawdopodobniej przez dużą co do modułu wartość pochodnej w stosunku do funkcji oryginalnej (około 10^4 razy większa w otoczeniu punktu $x = 0.01$).

4.9.2. Funkcje z osobliwościami, funkcje na przedziałach ograniczonych

Wyniki prezentowane w poprzednim podrozdziale są w dużym zakresie tożsame z wynikami uzyskiwanymi za pomocą biblioteki **chebfun** [120]. Jedyna zauważalna różnica dotyczy implementacji funkcji **eig** wykorzystywanej do obliczania pierwiastków. Implementacja tej metody w programie **Matlab** działa wyraźnie szybciej od funkcji dostępnej w bibliotece **numpy**. Dokładność, jak i czas obliczeń są zbliżone.

Rozwiązania przyjęte w pakiecie **PaCAL** w zakresie interpolacji funkcji określonych na przedziałach nieskończonych oraz funkcji w otoczeniu punktów osobliwych wykorzystują transformacje zmiennych i różnią się znacznie od rozwiązań stosowanych w pakiecie **chebfun**. W bibliotece **chebfun** jest stosowane adaptacyjne dzielenie przedziału na coraz mniejsze podprzedziały w otoczeniu osobliwości (lub coraz większe w przypadku przedziałów nieskończonych) oraz ekstrapolacja funkcją potęgową. Dla pewnych rodzajów funkcji oraz pewnych zastosowań takie rozwiązanie jest wystarczające, jednak w niektórych przypadkach (jak na przykład przy całkowaniu) uzyskiwana dokładność jest niska.

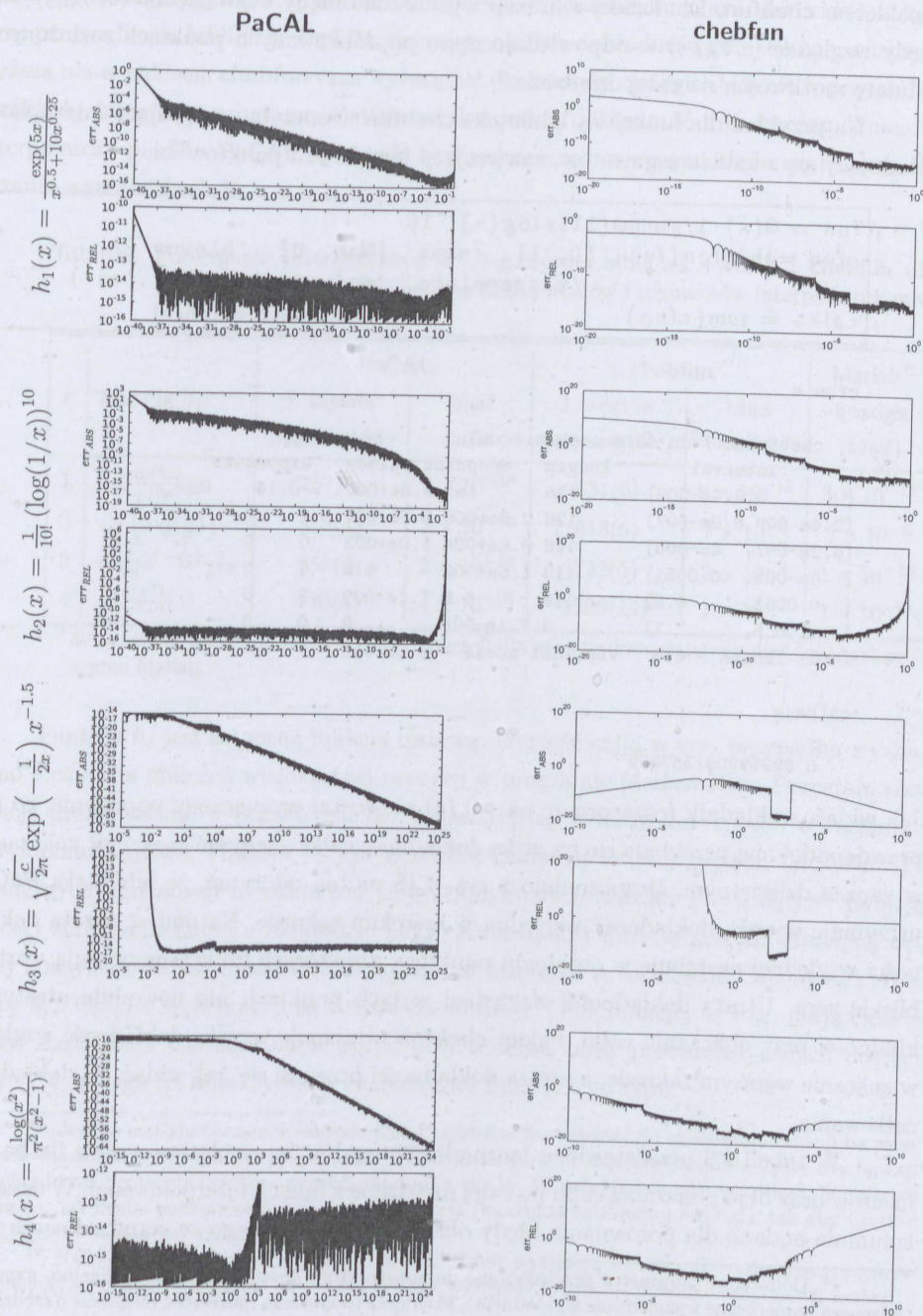
Przykład poniżej ilustruje efektywność rozwiązań stosowanych w pakiecie **PaCAL** w porównaniu z pakietem **chebfun**.

Przykład 4.7 (funkcje z biegunami). Rozważmy następujące funkcje:

$$\begin{aligned} - h_1(x) &= \frac{\exp(5x)}{x^{0.5} + 10x^{0.25}}, \quad x \in (0, 1], \\ - h_2(x) &= \frac{1}{10!} (\log(1/x))^{10}, \quad x \in (0, 1], \\ - h_3(x) &= \sqrt{\frac{c}{2\pi}} \exp\left(-\frac{1}{2x}\right) x^{-1.5}, \quad x \in (0, \infty), \\ - h_4(x) &= \frac{\log(x^2)}{\pi^2(x^2-1)}, \quad x \in (-\infty, 0) \cup (0, \infty). \end{aligned}$$

Trzy funkcje (h_1 , h_2 , h_4) mają w zerze punkty osobliwe. Funkcja h_1 zachowuje się asymptotycznie, jak funkcja potęgowa ($h_1 \sim x^{-0.25}$, $x \rightarrow 0$)¹⁵, natomiast funkcje h_2 i h_4 nie mogą być aproksymowane funkcją potęgową w otoczeniu zera ($h_2(x) \not\sim x^\alpha$, $x \rightarrow 0$ dla każdego α). Dwie ostatnie funkcje, h_3 i h_4 , są określone na przedziałach nieskończonych. Dla wszystkich funkcji utworzono interpolatory. Dla dwóch pierwszych funkcji zbudowano interpolator składający się z pojedynczego segmentu określonego na przedziale $(0, 1]$. W przypadku funkcji f_3 interpolator składał się z dwóch segmentów, określonych, odpowiednio, na przedziałach $(0, 1]$ oraz $[1, \infty)$. W przypadku funkcji f_4 wykorzystano cztery segmenty: $(-\infty, -1]$, $[-1, 0)$, $(0, 1]$ oraz $[1, \infty)$. W otoczeniu punktów osobliwych oraz w otoczeniu nieskończoności wykorzystano opisane transformacje zmiennych (potęgowe i logarytmiczne). Na rysunku 4.15 przedstawiono wyniki interpolacji w porównaniu

¹⁵ $f(x) \sim g(x)$, $x \rightarrow x_0$ oznacza $f(x)/g(x) \rightarrow 1$ dla $x \rightarrow x_0$.



Rys. 4.15. Porównanie bibliotek **PaCAL** i **chebfun** dla funkcji z przykładu 4.7; przedstawiono błędy interpolacji: bezwzględny err_{ABS} i względny err_{REL}

z pakietem **chebfun**. Dla każdej z funkcji wyznaczono błędy bezwzględne (err_{ABS}) oraz błędy względne (err_{REL}) w odpowiednio gęsto popróbkiowanych punktach rozłożonych, w miarę możliwości, na całej dziedzinie.

Na przykład dla funkcji h_2 biblioteka **chebfun** utworzyła następujący interpolator składający się z sześciu segmentów, zawierający łącznie 613 punktów¹⁶:

```
1 fun = @(x) 1/gamma(11)*log(x).^10
2 cfun =chebfun(fun, [0, 1], 'exp', [NaN, 0], 'blowup', 2, ...
3           'extrapolate', 'on', 'splitting', 'on')
4 calka = sum(cfun)
```

cfun =

chebfun column (6 smooth pieces)

interval	length	endpoint values	exponents
[0, 2.4e-009]	126	Inf 2.6e+006	-0.14 0
[2.4e-009, 6.2e-007]	126	2.6e+006 9.8e+004	0 0
[6.2e-007, 8e-005]	126	9.8e+004 1.5e+003	0 0
[8e-005, 0.0051]	113	1.5e+003 4.6	0 0
[0.0051, 0.5]	119	4.6 7.1e-009	0 0
[0.5, 1]	3	7.1e-009 0	0 0
Total length = 613		vertical scale = 2.6e+006	

calka =

0.999992097357329

Jak widać, wykładnik (oszacowany na -0.14) nie został wyznaczony poprawnie, co najprawdopodobniej przekłada się na niską dokładność całki – jedynie sześć cyfr znaczących w zapisie dziesiętnym. Bezpośrednio z rys. 4.15 można odczytać, że biblioteka **PaCAL** utrzymuje wysoką dokładność względną w szerokim zakresie. Natomiast utrata dokładności względnej następuje w otoczeniu punktów, dla których funkcje przyjmują wartości bliskie zera. Utrata dokładności względnej w tych punktach nie powoduje utraty dokładności przy obliczaniu całki. Pakiet **chebfun** utrzymuje wysoką dokładność względną w znacznie węższym zakresie, a utrata dokładności przenosi się, jak widać, na dokładność całkowania.

W tabeli 4.6 przedstawiono łączną liczbę węzłów interpolatora wraz z liczbą segmentów oraz błąd obliczania całki po całej dziedzinie z funkcji interpolowanej. W ostatniej kolumnie podano dla porównania błędy obliczeń dla najlepszego ze standardowych inte-

¹⁶ Dodatkowe parametry przekazywane do konstruktora **chebfun** oznaczają kolejno: **exp**=NaN (wymusza numeryczne wyznaczanie wykładnika), **blowup**=2 (dopuszcza możliwość biegunów o rzędzie niebędącym liczbą całkowitą, wartość 0 dla funkcji ograniczonych, 1 biegun całkowitego rzędu [74, rozdz. VI, VII]), **extrapolate**=on (wymusza ekstrapolację na końcach przedziału), **splitting**=on (pozwala na adaptacyjny podział na podsegmenty). Funkcja **sum** liczy całkę przez analogię tego polecenia dla wektora.

gratorów (quadgk¹⁷) programu Matlab całkujących oryginalną funkcję¹⁸. Jak widać, w każdym przypadku w bibliotece **PaCAL** dostajemy pełną dokładność numeryczną, znacznie wyższą niż w pakiecie **chebfun** oraz wyższą niż dla metody quadgk. Jedyne w przypadku funkcji h_3 integrator programu Matlab uzyskał pełną dokładność numeryczną. Ponadto interpolatory pakietu **PaCAL** są zwykle bardziej oszczędne, co przekłada się na czas obliczania wartości funkcji.

Tab. 4.6. Porównanie interpolatorów i integratorów bibliotek **PaCAL** i **chebfun** dla funkcji z przykładu 4.7 (podano liczbę węzłów i segmentów interpolatora oraz błąd obliczenia całki)

i	funkcja h_i	PaCAL		chebfun		Matlab*
		l. węzłów (segmentów)	błąd całki	l. węzłów (segmentów)	błąd całki	quadgk (błąd)
1	$\frac{\exp(5x)}{x^{0.5} + 10x^{0.25}}$	257(1)	4×10^{-15}	631(6)	8×10^{-13}	3×10^{-12}
2	$\frac{1}{10!} \left(\log\left(\frac{1}{x}\right) \right)^{10}$	512(1)	3×10^{-16}	613(6)	7×10^{-6}	3×10^{-8}
3	$\sqrt{\frac{1}{2\pi}} e^{(-\frac{1}{2x})} \frac{1}{x^{1.5}}$	258(2)	2×10^{-16}	132(5)	2×10^{-2}	1×10^{-16}
4	$\frac{\log(x^2)}{\pi^2(x^2-1)}$	770(4)	4×10^{-16}	1595(14)	1×10^{-4}	1×10^{-11}

* W ostatniej kolumnie podano błędy obliczeń dla standardowego integratora (quadgk) programu Matlab.

Funkcja h_1 jest sztuczną funkcją testową. Wartość całki w tym przypadku wyznaczono z użyciem obliczeń wielokrotnej precyzji w programie Mathematica. Pozostałe trzy funkcje prezentowane w przykładzie 4.7 pojawiają się w naturalny sposób w rachunku prawdopodobieństwa. Funkcja h_2 jest gęstością zmiennej losowej będącej iloczynem 11 niezależnych zmiennych o rozkładzie jednostajnym na przedziale $[0, 1]$, h_3 jest funkcją gęstości zmiennej losowej o rozkładzie Levy'ego, a h_4 jest funkcją gęstości zmiennej będącej iloczynem dwóch niezależnych zmiennych losowych o rozkładzie Cauchy'ego [113, rozdz. 4]). Całki z tych funkcji są oczywiście równe 1. Obie funkcje, h_3 i h_4 , mają ciężkie ogony. Zasadnicza trudność numeryczna przy liczeniu całek probabilistycznych postaci (2.1)–(2.4) polega na aproksymacji i całkowaniu funkcji tego typu.

¹⁷ Jest to metoda Gaussa–Kronroda [22], przystosowana również do całkowania funkcji na przedziałach nieskończonych oraz mających osłowności. Procedura ta działała znacznie lepiej dla funkcji z przykładu niż pozostałe dostępne procedury: quad i quadl. W przykładzie tym ustawiono dodatkowe parametry, jak liczba podziałów oraz zadana tolerancja (MaxIntervalCount i RelTol), tak aby uzyskać jak najwyższą dokładność obliczeń.

¹⁸ Należy zwrócić uwagę, że **PaCAL** oraz **chebfun** wykonują tu zadanie trudniejsze: najpierw interpolację, a później całkowanie; natomiast quadgk wykonuje jedynie całkowanie oryginalnej funkcji.

4.9.3. Funkcje z ciężkimi ogonami

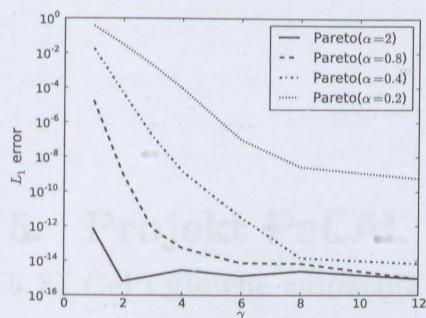
Jak wspomniano, funkcje z ciężkimi ogonami¹⁹ mogą powodować trudności numeryczne. Bierze się to z faktu, że istotna część masy probabilistycznej takich rozkładów jest rozłożona na dość szerokim zakresie. Jak wspomniano w podrozdz. 4.4.2 oraz 4.6.2, stosując transformacje zmiennych zbudowane na bazie funkcji potęgowych, należy podnieść wykładnik przekształcenia γ (w równaniu (4.23)), tak aby $\gamma > 1/\alpha$. Widać, że dla $\alpha > 1$ wartość wykładnika γ może być równa 1. Taka wartość jest teoretycznie wystarczająca dla wszystkich rozkładów mających przynajmniej pierwszy moment. Natomiast dla rozkładu Cauchy'ego oraz rozkładów z cięższymi ogonami (mających wykładniki ogonkowe w przedziale $[1, 2]$) konieczne jest zwiększenie wartości wykładnika. Następujący przykład numeryczny pokazuje wpływ wartości wykładnika γ na dokładność obliczeń przy wyznaczaniu spłotów funkcji gęstości rozkładów Pareto z różnymi wartościami parametrów:

Przykład 4.8 (spłot rozkładów Pareto). Rozważmy funkcję gęstości rozkładu Pareto:

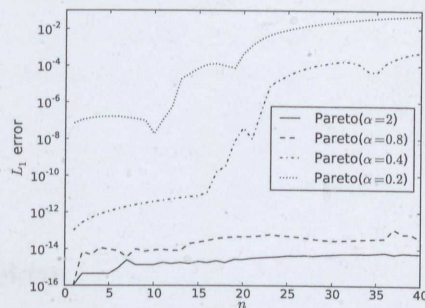
$$f_{\alpha}(x) = \alpha x^{-(1+\alpha)}, \quad \alpha > 0, \quad x \in [1, \infty).$$

Dla ustalonych wartości parametru $\alpha = 0.2, 0.4, 0.8, 2.0$ wyznaczono spłot 10 funkcji f_{α} dla różnych wartości wykładnika $\gamma = 1, 2, 3, 4, 6, 8, 12$ użytego przy transformacji zmiennych. Ponieważ dokładna teoretyczna postać funkcji gęstości jest trudna do wyznaczenia, jako miarę dokładności obliczeń przyjęto różnicę pomiędzy normą L^1 wyniku a wartością teoretyczną równą 1. Wyniki przedstawiono na rys. 4.16. Wzrost wartości wykładnika γ powoduje, że coraz szersza klasa funkcji uzyskuje wysoką dokładność obliczeń. Na przykład wartość wykładnika $\gamma = 8$ zapewnia pełną numeryczną dokładność obliczeń dla funkcji z parametrem $\alpha = 0.4, 0.8, 2.0$. Teoretycznie wartość ta powinna również zagwarantować dokładność dla funkcji z parametrem $\alpha = 0.2$ (ponieważ $8 > 1/0.2$), jednak w tym przypadku dostajemy dokładność jedynie na poziomie 1×10^{-9} . W kolejnym eksperymencie dla kolejnych wartości parametru $\alpha = 0.2, 0.4, 0.8, 2.0$ oraz dla ustalonego wykładnika transformacji $\gamma = 6$ wyznaczono spłot 40 funkcji f_{α} . Dokładności kolejnych wyników pośrednich w normie L^1 przedstawiono na rys. 4.17. Widać, że w trakcie kolejnych obliczeń błędy obliczeń narastają. W przypadku funkcji z parametrami $\alpha = 0.8, 2.0$ utrata dokładności jest niewielka, pod koniec obliczeń mamy w dalszym ciągu dokładność rzędu 1×10^{-14} . Natomiast dla wartości $\alpha = 0.2, 0.4$ po około 10, 15 iteracjach (operacjach spłotu) następuje znaczna utrata dokładności.

¹⁹ Zwykle definiuje się funkcje z ciężkimi ogonami jako takie, dla których pole pod „ogonem” maleje do zera wolniej niż wykładniczo ($e^{Ax} \int_x^{\infty} f(x) dx \rightarrow \infty$ dla $x \rightarrow \infty$ oraz dla wszystkich $A > 0$). Funkcje, o których mowa w tym akapicie, mają nawet cięższe ogony – takie, że $1 - F(x) \sim x^{-\alpha}$ dla $x \rightarrow \infty$, $0 \leq \alpha \leq 1$; wartość $-1-\alpha$ jest zwana wykładnikiem w nieskończoności lub wykładnikiem ogonkowym.



Rys. 4.16. Wpływ wykładnika γ przy transformacji zmiennych na dokładność obliczeń spłotu 10 funkcji gęstości Pareto z różnymi wartościami parametru α



Rys. 4.17. Spłot 40 funkcji gęstości rozkładu Pareto dla różnych wartości α , przedstawiono błędy w normie L^1 wyznaczone w kolejnych iteracjach ($\gamma = 6$)

Warto zaznaczyć, że w trakcie liczenia spłotów konieczne jest zachowanie wysokiej dokładności zarówno przy całkowaniu, jak i interpolacji, gdyż kolejne wyniki pośrednie muszą być interpolowane. Dokładność prezentowana w przykładzie 4.8 jest wypadkową dokładności interpolatorów, jak i integratorów.

Na koniec należy dodać uwagę na temat dokładności wyników przedstawionych w przykładzie 4.8. Bezpośrednio widać, że przy odpowiednio dużych wartościach wykładnika γ transformacji potęgowej obliczenia mogą być prowadzone z odpowiednio wysoką dokładnością. Za utratę dokładności na rys. 4.17 odpowiada częściowo niska wartość wykładnika, ale również domyślna graniczna liczba węzłów interpolacji (ustawiona w tym eksperymencie na 100 dla każdego segmentu interpolatora). Widać też, że dla funkcji f_α (z przykładu 4.8, rys. 4.16) z wartościami parametru α mniejszymi niż 0.4 dokładność obliczeń nie osiąga pełnej dokładności numerycznej nawet w początkowej fazie obliczeń, a jedynie 1×10^{-9} . Dyskusyjną sprawą jest to, czy jest to dokładność wysoka, biorąc pod uwagę fakt, że integrator `quadgk` programu `Matlab` nie jest w stanie wyznaczyć samej tylko całki $\int_1^\infty x^{-1.2} dx$ z dokładnością²⁰ wyższą niż 1×10^{-6} – przy wykładniku $\alpha = 0.4$ dostajemy maksymalnie dokładność rzędu 1×10^{-11} .

Dla parametrów $\alpha > 1$ oraz dla rozkładów szybko zbieżnych do zera problemy tego typu nie występują. W pracy [51] przedstawiono stosowne przykłady obliczeń spłotów dla rozkładów stabilnych (normalnego i Cauchy'ego); jest tam uzyskiwana pełna dokładność numeryczna.

²⁰ Aby uzyskać taką dokładność, należy zmienić domyślne parametry integratora, jak `RelTol` oraz `MaxIntervalCount`. Procedury całkowania lepszej jakości można znaleźć w pakiecie **QUADPACK** [99], por. też funkcję `quad` pakietu `scipy.integrate`.

5. Projekt PaCAL

5.1. Cel i główne założenia projektu

Głównym celem projektu **PaCAL** jest dostarczenie reprezentacji numerycznych oraz metod obliczeniowych do obliczeń probabilistycznych. Projekt składa się z dwóch głównych części. Pierwsza część dotyczy obliczeń na zmiennych losowych niezależnych, a druga wnioskowania probabilistycznego w ogólności na zmiennych zależnych.

Pomysłodawcami i wykonawcami projektu są Szymon Jaroszewicz i Marcin Korzeń. Projekt jest obecnie utrzymywany na portalu SourceForge (<http://sourceforge.net/pacal>). Strona domowa projektu to <http://pacal.sf.net>. Biblioteka została napisana w języku Python (wersje 2.6.x lub 2.7.x) [103, 122]. Wybór tego środowiska jest podyktowany tym, że jest to środowisko w pełni darmowe oraz zapewnia wsparcie odpowiedniej jakości dla niskopoziomowych operacji numerycznych, jak działania na wektorach i macierzach (biblioteka **numpy** [4]).

Jednym z głównych założeń projektowych było wykonywanie obliczeń w sposób numerycznie dokładny, z wykorzystaniem odpowiednich reprezentacji numerycznych. Dzięki takiemu podejściu możliwe okazało się osiągnięcie jednocześnie wysokiej dokładności oraz wydajności obliczeń. Takie podejście różni się zasadniczo od powszechnie stosowanych rozwiązań w obliczeniach probabilistycznych, które zwykle dotyczą przypadków szczególnych lub opierają się na metodach typu Monte Carlo. Dodatkowo pakiet miał umożliwiać obliczenia na możliwie szerokiej klasie zmiennych losowych, w tym również obliczenia na zmiennych niezależnych, jak i zależnych.

W pakiecie **PaCAL** możliwe są obliczenia na zmiennych losowych niezależnych dowolnych typów: ciągłych, dyskretnych, mieszanych. Natomiast przypadek zmiennych zależnych, ograniczony do zmiennych ciągłych, oraz zmienne na przedziałach nieskończonych są obecnie obsługiwane tylko w niektórych typach obliczeń. Pewną liczbę rozkładów, zarówno jednowymiarowych, jak i wielowymiarowych, zdefiniowano w sposób natywny. Rozkłady innych zmiennych losowych można tworzyć, wykonując na nich dalsze operacje. Dla zmiennych niezależnych zaimplementowano cztery podstawowe działania arytmetyczne (dodawanie, odejmowanie, mnożenie, dzielenie), operacje minimum, maksimum oraz potęgowanie zmiennych losowych. Zdefiniowano również szereg funkcji dla zmiennych losowych, jak potęga, funkcja wykładnicza czy logarytm. Możliwe jest generowanie liczb

losowych o dowolnym rozkładzie, rysowanie histogramów, wykresów, wyznaczanie statystyk podsumowujących, jak średnia, mediana czy przedziały ufności. Część dotycząca zmiennych zależnych umożliwia wnioskowanie probabilistyczne obejmujące warunkowanie, marginalizację czy obliczenia na zmiennych zależnych. Stan biblioteki **PaCAL** w wersji 1.1 od strony użytkownika został opisany w pracy [68], która tematycznie pokrywa się z treścią tego rozdziału.

5.2. Instalacja i pierwsze uruchomienie

Biblioteka **PaCAL** jest zależna od następujących bibliotek zewnętrznych:

NumPy – główna biblioteka zewnętrzna zapewniająca niskopoziomowe operacje na wektorach i macierzach ($\geq 1.4.1$ [4]).

Matplotlib – standardowa biblioteka dostarczająca funkcje graficzne oraz umożliwiającą prezentację wyników ($\geq 1.0.0$, [47]).

Sympy – pakiet obliczeń symbolicznych wymagany w wersji **PaCAL** ≥ 1.1 , wykorzystywany do symbolicznej zamiany zmiennych w przypadku obliczeń na zmiennych zależnych; wersja biblioteki **PaCAL** 1.0, która zawiera jedynie arytmetykę zmiennych niezależnych, nie wymaga tego pakietu ($\geq 0.6.7$ [117]).

SciPy – rozbudowana biblioteka do obliczeń numerycznych, wykorzystywana w pewnym zakresie do znajdowania maksimów i rozwiązywania równań (przy odwracaniu dystrybuanty), pakiet docelowo opcjonalny (≥ 0.9 [61]).

Cython – biblioteka umożliwiająca kompilacje fragmentów kodu napisanych w języku C/C++, co w pewnym stopniu przyspiesza obliczenia, pakiet opcjonalny [6].

Dla użytkowników systemu Windows dostarczono wykonywalną wersję instalacyjną, w przeciwnym razie należy pobrać plik rozpakować `tar.gz`, a następnie zainstalować za pomocą standardowego polecenia:

```
python setup.py install
```

Wersję źródłową (najbardziej aktualną) biblioteki można pobrać bezpośrednio z publicznego repozytorium GIT¹:

```
git clone git://git.code.sf.net/p/pacal/code pacal-code
```

lub SVN²:

```
svn checkout svn://svn.code.sf.net/p/pacal/svn/trunk pacal-svn
```

Następujący fragment kodu³ inicjuje zmienną o standaryzowanym rozkładzie normalnym, wyświetla statystyki podsumowujące oraz rysuje wykres funkcji gęstości:

¹ <http://git-scm.com/>, dostęp 1.06.2012 r.

² <http://subversion.apache.org/>, dostęp 1.06.2012 r.

³ Pierwsza linia importuje publiczne klasy biblioteki **PaCAL**. W dalszej części linie importujące oraz fragmenty kodu formatujące wykresy zostały pominięte, jednak po zaimportowaniu biblioteki

```

1 from pacal import *
2
3 N = NormalDistr()
4 N.summary(show_moments=True)
5 N.plot()
6 show()

```

```
===== summary =====
```

```
N(0.0,1.0)
```

```

      mean = 2.2204460492503131e-16
      var  = 1.00000000000000013
      skewness = -5.5511151231257719e-16
      kurtosis = 2.9999999999999947
      entropy = 1.4189385332046736
      median = 1.1102230246251565e-16
      mode   = -1.9750065193039777e-16
      medianad = 0.6744897501960747
      iqrang(0.025) = 3.9199279690798607
      ci(0.05) = (-1.9599639845398045, 1.9599639845400563)
      range   = (-inf, inf)
      tailexp  = (-235.71613043125413, -235.71613043125413)
      int_err  = 1.1102230246251565e-16

```

```
moments:
```

```

0 = 0.99999999999999989
1 = 2.2204460492503131e-16
2 = 1.00000000000000011
3 = 1.1102230246251565e-16
4 = 3.00000000000000022
5 = 0.0
6 = 15.0000000000000018
7 = 0.0
8 = 104.999999999999991
9 = 0.0
10 = 945.00000000000011

```

W tym przypadku rozkład normalny ma zdefiniowaną wcześniej funkcję gęstości, jednak wszystkie prezentowane wskaźniki są wyznaczone w sposób numeryczny. Jak widać, dokładność obliczeń jest wysoka nawet dla momentów wyższych rzędów. Podobną dokładność uzyskuje się również w bardziej złożonych operacjach.

5.3. Obliczenia na zmiennych niezależnych

5.3.1. Szczegóły implementacyjne

Biblioteka składa się z następujących pakietów:

pacal – główny pakiet implementujący obliczenia na zmiennych losowych niezależnych;

wszystkie przykłady mogą być uruchomione metodą kopiuj-wklej. Polecenie `show()` jest komendą biblioteki **matplotlib**, powoduje wyświetlenie okna graficznego z wykresami.

`pacal.depvars` – pakiet implementujący obliczenia probabilistyczne na zmiennych losowych zależnych;

`pacal.stats` – pakiet zawierający proste wsparcie dla metod wnioskowania statystycznego w postaci działań na zbiorach niezależnych zmiennych losowych o jednakowym rozkładzie (jak suma średnia, iloczyn czy średnia geometryczna) oraz metody do estymacji parametrów rozkładów na podstawie próbek;

`pacal.examples` – zestaw przykładów różnego typu prezentujących możliwości biblioteki oraz służących do celów testowych.

Reprezentacje zmiennych losowych niezależnych dziedziczą po abstrakcyjnej klasie `Distr`. Klasa ta ma dwie metody abstrakcyjne:

`init_pieewise_pdf()` – inicjuje funkcję gęstości rozkładu, inicjowane jest pole `pieewise_pdf` tej klasy.

`rand_raw(n)` – definiuje generator liczb losowych.

Obiekty klasy `Distr` reprezentują zmienne losowe, a numeryczną reprezentacją zmiennej losowej jest funkcja gęstości prawdopodobieństwa (zdefiniowana za pomocą obiektu `PiecewiseDistribution`). Wszystkie dalsze obliczenia są wykonywane na obiektach klasy `Distr` oraz opierają się na założeniu niezależności. Wyniki różnych operacji probabilistycznych na zmiennych losowych są ponownie instancjami klasy `Distr` i mogą brać udział w dalszych obliczeniach. Główna część pakietu (moduł `indeparith`) implementuje cztery operacje arytmetyczne na zmiennych losowych niezależnych. Dla obiektów dziedziczących po klasie `Distr` przeciążono operatory: `+`, `-`, `*`, `/`. Ponadto zaimplementowano operacje minimum, maksimum (funkcje `min`, `max`) oraz potęgę (operator `**`). Wszystkie operacje binarne są określone zarówno dla przypadku, gdy oba argumenty są zmiennymi losowymi, jak również dla przypadku, gdy jeden z argumentów jest liczbą zmiennopozycyjną. Zaimplementowano również pewną liczbę funkcji zmiennych losowych.

W klasie `Distr` zdefiniowano wiele metod umożliwiających rysowanie wykresów, generowanie liczb losowych, wyznaczanie dystrybuant czy statystyk podsumowujących. Obliczenia mogą być wykonywane zarówno na zmiennych ciągłych, jak i dyskretnych w jednolity sposób, oba rodzaje zmiennych mogą być mieszane ze sobą.

5.3.2. Rozkłady jednowymiarowe zdefiniowane w **PaCAL**

Poniżej przedstawiono standardowe rozkłady jednowymiarowe zebrane w module `standard.distr` (w nawiasach podano parametry domyślne):

`UniformDistr(a=0.0, b=1.0)` – zmienna o rozkładzie jednostajnym $U(a, b)$, funkcja gęstości dana wzorem:

$$f(x; a, b) = \begin{cases} \frac{1}{b-a}, & x \in [a, b], \\ 0, & x \notin [a, b]; \end{cases}$$

`TrapezoidalDistr(a=0.0, b=0.0, c=1.0, d=1.0)` – zmienna o rozkładzie trapezowym $Trapz(a, b, c, d)$, funkcja gęstości dana wzorem:

$$f(x; a, b, c, d) = \begin{cases} u \frac{x-a}{b-a}, & x \in [a, b], \\ u, & x \in [b, c], \\ u \frac{d-x}{d-c}, & x \in [c, d], \\ 0 & \text{w pozostałych przypadkach,} \end{cases}$$

gdzie $a \leq b \leq c \leq d$ oraz $u = \frac{2}{d+c-a-b}$;

`BetaDistr(alpha=1, beta=1)` – zmienna o rozkładzie beta $Beta(\alpha, \beta)$, funkcja gęstości dana wzorem:

$$f(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1};$$

`NormalDistr(mu=0.0, sigma=1.0)` – zmienna o rozkładzie normalnym $N(\mu, \sigma)$, funkcja gęstości dana wzorem:

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

`CauchyDistr(gamma=1.0, center=0.0)` – zmienna o rozkładzie Cauchy'ego $C(\gamma, c)$, funkcja gęstości dana wzorem:

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

`StudentTDistr(df=2)` – zmienna o rozkładzie t -Studenta $T(df)$, funkcja gęstości dana wzorem:

$$f(t) = \frac{\Gamma(\frac{df+1}{2})}{\sqrt{df\pi} \Gamma(\frac{df}{2})} \left(1 + \frac{t^2}{df}\right)^{-\frac{df+1}{2}};$$

`FDistr(df1=1, df2=1)` – zmienna o rozkładzie F Snedecora $F(df_1, df_2)$, funkcja gęstości dana wzorem:

$$f(x; df_1, df_2) = \frac{1}{B\left(\frac{df_1}{2}, \frac{df_2}{2}\right)} \left(\frac{df_1}{df_2}\right)^{\frac{df_1}{2}} x^{\frac{df_1}{2}-1} \left(1 + \frac{df_1}{df_2} x\right)^{-\frac{df_1+df_2}{2}};$$

`ChiSquareDistr(df=1)` – zmienna o rozkładzie χ^2 $Chi2(df)$, funkcja gęstości dana wzorem:

$$f(x; df) = \begin{cases} \frac{1}{2^{df/2} \Gamma(df/2)} x^{df/2-1} e^{-x/2}, & x \in [0, \infty), \\ 0, & x \in (-\infty, 0); \end{cases}$$

`GammaDistr(k=2, theta=2)` – zmienna o rozkładzie gamma $Gamma(k, \theta)$, funkcja gęstości dana wzorem:

$$f(x; k, \theta) = \begin{cases} \frac{1}{\theta^k} \frac{1}{\Gamma(k)} x^{k-1} e^{-x/\theta}, & x \in [0, \infty), \\ 0, & x \in (-\infty, 0); \end{cases}$$

`ExponentialDistr(lmbda=1)` – zmienna o rozkładzie wykładniczym $E(\lambda)$, funkcja gęstości dana wzorem:

$$f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x}, & x \in [0, \infty), \\ 0, & x \in (-\infty, 0); \end{cases}$$

`LaplaceDistr(lmbda=1.0, mu=0.0)` – zmienna o rozkładzie Laplace’a $L(\lambda, \mu)$, funkcja gęstości dana wzorem:

$$f(x|\mu, \lambda) = \frac{1}{2\lambda} \exp\left(-\frac{|x - \mu|}{\lambda}\right);$$

`ParetoDistr(alpha=1, xmin=1)` – zmienna o rozkładzie Pareto $P(\alpha, x_{\min})$, funkcja gęstości dana wzorem:

$$f(x; x_{\min}, \alpha) = \begin{cases} \alpha \frac{x_{\min}^\alpha}{x^{\alpha+1}}, & x \in [x_{\min}, \infty), \\ 0, & x \in (-\infty, x_{\min}); \end{cases}$$

`LevyDistr(c=1.0, xmin=0.0)` – zmienna o rozkładzie Levy’ego $L(c, x_{\min})$, funkcja gęstości dana wzorem:

$$f(x; x_{\min}, c) = \sqrt{\frac{c}{2\pi}} (x - x_{\min})^{-\frac{3}{2}} \exp\left(-\frac{c}{2(x - x_{\min})}\right);$$

`WeibullDistr(k=3.0, lambda=1.0)` – zmienna o rozkładzie Weibulla $We(k, \lambda)$, funkcja gęstości dana wzorem:

$$f(x; \lambda, k) = \begin{cases} \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}, & x \in [0, \infty), \\ 0, & x \in (-\infty, 0); \end{cases}$$

`GumbelDistr(mu=0, sigma=1)` – zmienna o rozkładzie Gumbela $Gu(\mu, \sigma)$, funkcja gęstości dana wzorem:

$$f(x; \mu) = \frac{1}{\sigma} \exp\left(-\frac{x-\mu}{\sigma} - \exp\left(-\frac{x-\mu}{\sigma}\right)\right), \quad x \in (-\infty, \infty);$$

`FrechetDistr(alpha=2, s=1, m=0)` – zmienna o rozkładzie Frécheta $Fr(\alpha, s, m)$, funkcja gęstości dana wzorem:

$$f(x; s, m) = \begin{cases} \frac{\alpha}{s} \left(\frac{x-m}{s}\right)^{-1-\alpha} \cdot e^{-(\frac{x-m}{s})^{-\alpha}}, & x \in [m, \infty), \\ 0, & x \in (-\infty, m). \end{cases}$$

Te i inne rozkłady oraz wzajemne relacje pomiędzy rozkładami można znaleźć w przeglądowej pracy [73] (patrz też [50, 130]). Bardziej złożone rozkłady uzyskujemy w wyniku wykonywania różnych operacji na zmiennych losowych. Na przykład w pakiecie `pacal.stats` zaimplementowano rozkłady niecentralne wykorzystywane w praktyce [113]:

`NoncentralTDistr(df=2, mu=0)` – niecentralny rozkład T zdefiniowany jako:

$$NCT(df, \mu) = \frac{N(\mu, 1)}{Chi2(df)};$$

`NoncentralChiSquareDistr(df, lambda=0)` – niecentralny rozkład chi-kwadrat (definiowany jako suma kwadratów rozkładów normalnych przesuniętych względem zera i dzielonych przez wariancje [113, podrozdz. 9.2]):

$$NCX(df, \lambda) = \sum_{i=1}^{df} \left(\frac{N(\lambda, \sigma_i)}{\sigma_i} \right)^2 = Chi2(df-1) + N(\sqrt{\lambda}, 1);$$

druga równość zachodzi dla przypadku $\sigma_1 = \dots = \sigma_{df} = 1$;

`NoncentralFDistr(df1=1, df2=1, lambda=0)` – niecentralny rozkład F (zdefiniowany jako iloraz zmiennej o niecentralnym rozkładzie chi-kwadrat i zmiennej o centralnym

rozkładzie χ^2)⁴:

$$NCF(df_1, df_2) = \frac{NCX(df_1, \lambda)/df_1}{Chi2(df_2)/df_2},$$

`NoncentralBetaDistr(alpha=1, beta=1, lmbda=0)` – niecentralny rozkład beta zdefiniowany jako iloraz:

$$NCB(\alpha, \beta, \lambda) = \frac{X}{X+Y} \text{ dla } X \sim NCX(\alpha, \lambda), Y \sim NCX(\beta, \lambda).$$

Inne rozkłady mogą być definiowane przez użytkownika z wykorzystaniem dziedziczenia bezpośrednio po klasie `Distr` (wymaga implementacji metod abstrakcyjnych) lub z użyciem klasy:

`FunDistr(fun, breakPoints=None)`.

W tym przypadku należy jawnie podać funkcję gęstości (parametr `fun`) oraz listę punktów charakterystycznych dziedziny (parametr `breakPoints` określający początek i koniec dziedziny oraz punkty nieciągłości); mogą być tu używane symbole `-Inf`, `+Inf` do określenia przedziałów nieskończonych.

Rozkłady dyskretne w pakiecie **PaCAL** rozszerzają klasę `DiscreteDistr`, która jest podklasą klasy `Distr`. Konstruktor klasy `DiscreteDistr(xi, pi)` wymaga listy punktów (`xi`) oraz wartości prawdopodobieństw w określonych punktach (`pi`). Obecnie możliwe jest tworzenie jedynie rozkładów dyskretnych zmiennych losowych przyjmujących skończoną liczbę wartości. Dostępne rozkłady dyskretne to:

`ConstDistr(c=1.0)` – rozkład jednopunktowy; w ten sposób zmienne rzeczywiste „wkładają się”, jako przypadek szczególny, w zmienne losowe reprezentujące wartości pewne. Szczególnymi rozkładami są tu: `ZeroDistr()` (rozkład jednopunktowy skupiony w zerze, dystrybucja Diraca, element neutralny dodawania) oraz `OneDistr()` (rozkład jednopunktowy skupiony w jedności, element neutralny mnożenia).

`BernoulliDistr(p=0.5)` – rozkład Bernoulliego (dwupunktowy) $Bernoulli(p)$, określony na zbiorze $\{0, 1\}$, $\Pr(X = 1) = p$.

`BinomialDistr(n=2, p=0.5)` – rozkład dwumianowy $Binom(n, p)$, suma n niezależnych zmiennych o rozkładzie Bernoulliego z parametrem p .

Ponadto zaimplementowano mieszanekę rozkładów `MixDistr(pi, fi)` (średnia ważona funkcji gęstości), gdzie `fi` jest listą rozkładów, `pi` są wagami. Funkcja gęstości w tym przypadku jest określona jako: $f(x) = \sum_{i=1}^n p_i f_i(x)$.

⁴ Definiuje się również podwójnie niecentralny rozkład F jako iloraz dwóch niecentralnych rozkładów chi-kwadrat. Używając pakietu **PaCAL**, tego typu rozkłady wyznaczamy z łatwością.

5.3.3. Metody klasy Distr

Obiekty klasy `Distr` są wyposażone w szereg metod umożliwiających łatwą analizę uzyskanych wyników. W tabeli 5.1 zebrano szczegółowy opis dostępnych parametrów zmiennych losowych. Wszystkie parametry liczone są numerycznie, zwykle przez bezpośrednie całkowanie czy rozwiązywanie równań. Metoda `summary()` wyświetla podsumowanie podstawowych parametrów zmiennej losowej.

Tab. 5.1. Statystyki podsumowujące zdefiniowane w klasie `Distr`

Metoda	Opis
<code>mean()</code>	średnia, wartość oczekiwana liczona jako całka $\int_{-\infty}^{\infty} x f(x) dx$
<code>var()</code>	wariancja: $\int (x - m)^2 f(x) dx$
<code>std()</code>	odchylenie standardowe, pierwiastek kwadratowy z wariancji
<code>moment(k, c)</code>	k -ty moment zmiennej losowej względem wartości c ; gdy c jest pominięte, wyznaczone są momenty centralne (względem średniej)
<code>skewness()</code>	skośność: <code>moment(3, mean()) / var() ** 3</code>
<code>kurtosis()</code>	kurtoza: <code>moment(4, mean()) / var() ** 4</code>
<code>entropy()</code>	entropia zmiennej losowej: $-\int f(x) \log(f(x)) dx$
<code>quantile(level)</code>	kwantyl na poziomie <code>level</code> , rozwiązanie równania $\int_{-\infty}^y f(x) dx = \text{level}$
<code>median()</code>	mediana, kwantyl na poziomie $\frac{1}{2}$: <code>quantile(0.5)</code>
<code>medianad()</code>	bezwzględne rozproszenie medianowe: <code>median(X - median(X))</code>
<code>iqrang(level)</code>	rozstęp międzykwantylowy na poziomie <code>level</code> : <code>quantile(1-level) - quantile(level)</code>
<code>range()</code>	zakres (dziedzina) zmiennej losowej
<code>ci(p=0.05)</code>	symetryczny przedział ufności na poziomie istotności p : <code>(quantile(p/2), quantile(1-p/2))</code>
<code>tail_exponent()</code>	estymacja wykładnika (ogonkowego) w nieskończoności, wartość β w wyrażeniu $f(x) = \mathcal{O}(\frac{1}{x^\beta})$ dla $x \rightarrow \pm\infty$, wartości $\beta > 100$ należy interpretować jako tempo zbliżności podobne do wykładniczego
<code>mode()</code>	wartość modalna, ekstremum funkcji gęstości

Oprócz statystyk opisowych jest dostępnych szereg metod umożliwiających rysowanie wykresów, generowanie liczb losowych czy szacowanie błędów obliczeń (tab. 5.2). Wszystkie metody graficzne mają możliwość formatowania wykresów, przekazując do metod graficznych standardowe argumenty funkcji `plot` pakietu `matplotlib`.

Tab. 5.2. Pozostałe metody klasy `Distr`

Metoda	Opis
Funkcje graficzne	
<code>plot</code>	rysuje funkcję gęstości, parametry: <code>xmin</code> , <code>xmax</code> – zakresy
<code>hist</code>	rysuje histogram lub histogram warunkowy, parametry: <code>n</code> – liczba próbek, <code>xmin</code> , <code>xmax</code> – zakresy dla histogramów warunkowych, <code>bind</code> – liczba pudełek
<code>boxplot</code>	wykres pudełkowy, parametry: <code>pos</code> – pozycja, <code>width</code> – szerokość, <code>vertical</code> – pionowo/poziomo
Generowanie liczb losowych	
<code>rand(n)</code>	generuje liczby losowe, korzystając z metod natywnych
<code>rand.invcdf(n)</code>	generowanie liczb losowych metodą odwracania dystrybucji, dodatkowy parametr <code>use_interpolated=True</code> wymusza interpolację
Funkcja gęstości i dystrybucji	
<code>pdf(x)</code>	wartość funkcji gęstości w punkcie <code>x</code> (<code>x</code> może być wektorem wartości)
<code>cdf(x)</code>	wartość dystrybucji w punkcie <code>x</code> (<code>x</code> może być wektorem wartości)
<code>quantile(y)</code>	wartość funkcji odwrotnej do dystrybucji (kwantyl) w punkcie <code>y</code>
<code>get_piecewise_pdf()</code>	obiekt reprezentujący funkcję gęstości
<code>get_piecewise_cdf()</code>	obiekt reprezentujący dystrybucję
<code>get_piecewise_invcdf()</code>	obiekt reprezentujący funkcję odwrotną do dystrybucji (funkcję kwantyli)
Testowanie dokładności	
<code>int_error()</code>	błąd obliczeń szacowany na podstawie normy L^1 : $\text{int_error} = 1 - \int_{-\infty}^{\infty} f(x) dx$
<code>interp_error.by_segment()</code>	szacowane błędy bezwzględne interpolatorów w poszczególnych segmentach

Metoda `int_error` zwraca błąd w normie L^1 – błąd całki z rozkładu względem wartości 1. Praktyka pokazuje, że wskaźnik ten bardzo dobrze szacuje dokładność obliczeń dla większości parametrów.

W pewnych przypadkach wygodne jest bezpośrednie posługiwanie się funkcją gęstości lub dystrybuantą jako obiektem klasy `PiecewiseFunction` (podrozdz. 4.8). Ilustruje to poniższy przykład:

```

1 S = UniformDistr() + UniformDistr()
2 S.plot()
3 S.get_pieewise_cdf().plot()
4 print S.get_pieewise_pdf().integrate(0.5,1.5)
5 print S.get_pieewise_cdf()(1.5) - S.get_pieewise_cdf()(0.5)
6 T = FunDistr(fun=lambda x: 1-abs(x-1), breakPoints=[0,1,2])
7 r = T.get_pieewise_pdf() - S.get_pieewise_pdf()
8 r.plot()
9 print sqrt((r**2).integrate())

```

0.75

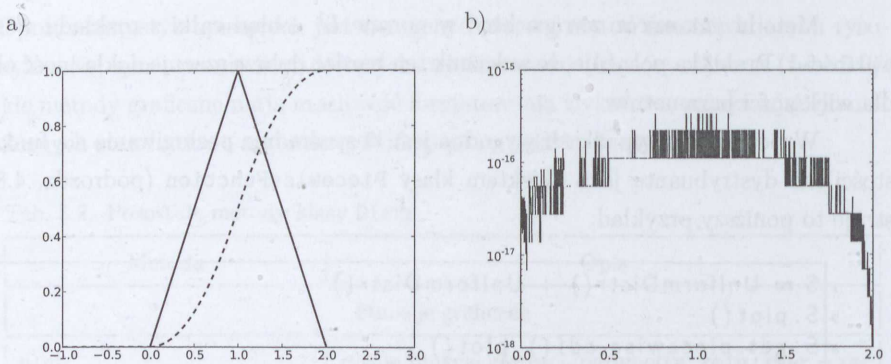
0.75

1.43474427472e-16

W przykładzie tym obliczono sumę dwóch niezależnych zmiennych losowych o rozkładzie jednostajnym, dostając w wyniku rozkład trójkątny. Następnie są rysowane wykresy funkcji gęstości oraz dystrybuanty. Kolejne linie (4,5) wyznaczają $\Pr(0.5 < S < 1.5)$ na dwa różne sposoby: bezpośrednim całkowaniem funkcji gęstości oraz jako różnicę dystrybuant. Dystrybuanta jest obliczana przez bezpośrednie całkowanie funkcji gęstości w wybranych węzłach oraz następnie z użyciem interpolacji na całej dziedzinie. Funkcjonalność związana z klasą `PiecewiseFunction` umożliwia wykonywanie operacji punktowych na funkcjach (jak suma czy iloczyn). W przykładzie tym porównano teoretyczny rozkład trójkątny (zmienna `T`) z rozkładem sumy (zmienna `S`). Na rysunku 5.1a przedstawiono wykresy dystrybuanty oraz funkcji gęstości zmiennej `S`, a na rys. 5.1b – punktową różnicę pomiędzy funkcją gęstości wyznaczoną numerycznie (zmienna `S`) oraz teoretycznym rozkładem trójkątnym (zmienna `T`). Ostatnia linia liczy normę $\|\cdot\|_2$ różnicy `r`, która – jak widać – jest na poziomie precyzji maszynowej. Błąd w normie L^∞ można odczytać bezpośrednio z wykresu.

5.3.4. Arytmetyka zmiennych losowych niezależnych

Operacje numeryczne na zmiennych losowych niezależnych reprezentowanych przez obiekty klasy `Distr` wykonuje się w taki sam sposób jak na zwykłych zmiennych numerycznych. Dopuszczalne są operatory binarne: `+`, `-`, `*`, `/`, potęgowanie `**`, `min`, `max` oraz funkcje: `exp`, `log`, `atan`, `sign`, `abs`, `sin`, `cos`, `tan`. Funkcje trygonometryczne są zdefiniowane jedynie dla zmiennych określonych na okresie danej funkcji trygonometrycznej.



Rys. 5.1. Suma dwóch zmiennych o rozkładzie jednostajnym: a) funkcja gęstości oraz dystrybuanta, b) różnica pomiędzy rozkładem wyznaczonym numerycznie S a rozkładem teoretycznym T

Operatory binarne mogą być stosowane również w sytuacjach, gdy jeden z argumentów jest zwykłą zmienną rzeczywistą (wtedy jest on traktowany jako rozkład jednopunktowy).

5.3.5. Zmienne dyskretne i mieszane

Zmienne dyskretne tworzymy, wykorzystując bezpośrednio klasę `DiscreteDistr` lub jej podklasy. Na zmiennych dyskretnych można wykonywać wszystkie operacje identycznie jak na zmiennych ciągłych. Zmienne takie mogą być mieszane ze sobą. Kolejny przykład przedstawia próbkę możliwości.

```

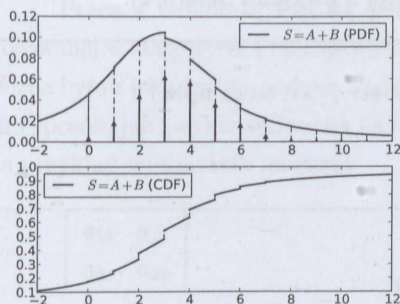
1 D = BinomialDistr(6, 0.3)
2 C = CauchyDistr(2, 1)
3 A = min(C, D)
4 B = max(C, D)
5 S = A + B
6 S.plot()
7
8 D = DiscreteDistr(xi=[1, 5, 8], pi=[0.2, 0.5, 0.3])
9 N = NormalDistr()
10 M = D + N
11 M.plot()
12 print M.mean()

```

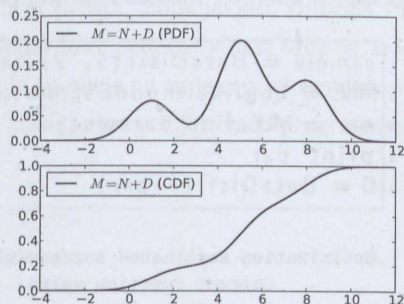
5.1000000000000005

W przykładzie tym są liczone kolejno minimum (A) i maksimum (B) zmiennej ciągłej o rozkładzie Cauchy'ego oraz zmiennej o rozkładzie dwumianowym. Następnie zmienne, traktowane jako zmienne niezależne, dodajemy do siebie (zmienna S). Wynik – rozkład i dystrybuanta zmiennej S – został przedstawiony na rys. 5.2. Widać część ciągłą, jak

i dyskretną. W drugiej części (linie 8–12) wyznaczono sumę zmiennej losowej o rozkładzie normalnym oraz zmiennej o rozkładzie dyskretnym (mieszanka rozkładu normalnego). W tym przypadku wynik – przedstawiony na rys. 5.3 – jest zmienną ciągłą (podrozdz. 2.6).



Rys. 5.2. Suma dwóch zmiennych mieszanym, rozkład zawiera część ciągłą i dyskretną



Rys. 5.3. Mieszanka rozkładów normalnych, suma zmiennej normalnej oraz zmiennej dyskretniej

5.3.6. Rozkłady warunkowe

W pakiecie **PaCAL** można definiować rozkłady warunkowe jednowymiarowe – rozkłady zmiennych odciętych do pewnego przedziału. Zdefiniowano dwie klasy:

`CondGtDistr(X, L)` – zmienna losowa X pod warunkiem $x > L$,

`CondLtDistr(X, U)` – zmienna losowa X pod warunkiem $x > U$,

gdzie U i L są stałymi. Aby uczynić notację bliższą notacji probabilistycznej przeciążono symbol `|` (operator *lub* języka Python). Składnia tworzenia rozkładów warunkowych jest następująca:

$X \mid \text{warunek}$,

gdzie X jest zmienną losową (instancją klasy `Distr`) oraz *warunek* jest jednym z: `Gt(L)`, `Lt(U)`, `Between(L, U)`. Poniższy fragment kodu pokazuje, że rozkład wykładniczy jest rozkładem bez pamięci, to znaczy po warunkowaniu i przesunięciu dostajemy ponownie identyczny rozkład wykładniczy.

```
1 E = ExponentialDistr()
2 CE = E | Gt(7)
3 r = E.get_pieewise_pdf() - (CE - 7).get_pieewise_pdf()
4 print "L1 norma =", r.max_abs()[1]
```

L1 norma = 4.3520742565306136e-13

5.3.7. Estymacja parametrów

Moduł `stats.distr_est` zawiera pojedynczą klasę `LoglikelihoodEstimator`, która może być wykorzystana do estymacji parametrów rozkładów zdefiniowanych w module `standard_distr`. Poniższy przykład ilustruje estymację parametrów rozkładu beta na podstawie losowej, 100-elementowej próbki wziętej z rozkładu $Beta(5, 2)$.

```
1 sample = BetaDistr(5, 2).rand(100)
2 MLE = LoglikelihoodEstimator(BetaDistr, xi = sample)
3 par = MLE.find_params()
4 print par
5 B = BetaDistr(**par)
```

```
Optimization terminated successfully.
      Current function value: -47.052775
      Iterations: 46
      Function evaluations: 88
{'alpha': 5.2116062279210915, 'beta': 2.4069282601561408}
```

5.4. Semantyka

Biblioteka **PaCAL** umożliwia wykonywanie obliczeń na zmiennych losowych w taki sposób, jak wykonuje się obliczenia na zwykłych zmiennych. Jednak bezpośrednie przeniesienie semantyki obliczeń na typach liczbowych na zmienne losowe wymaga pewnej uwagi. Zmienne losowe oznaczone tym samym symbolem powinny być traktowane jako identyczne (odróżnia się tu przypadek zmiennych o jednakowym rozkładzie, na przykład zmiennych niezależnych). Tego typu uwaga nie zawsze jest prawidłowo implementowana w pakietach umożliwiających obliczenia na zmiennych niepewnych, jednym z wyjątków jest tu biblioteka **uncertainties** [72], która prawidłowo obsługuje taki przypadek i jasno o tym informuje w dokumentacji. W pakiecie **PaCAL** występują dwie różne konwencje. W przypadku zmiennych niezależnych wszystkie zmienne biorące udział w operacjach na każdym etapie obliczeń są traktowane jako niezależne.

Ilustruje to poniższy fragment kodu:

```
1 S1 = UniformDistr() + UniformDistr()
2 U = UniformDistr(); S2 = U + U
```

```
Warning: arguments treated as independent
```

Wyniki obu operacji mają taki sam rozkład, jednak druga operacja powoduje wyświetlenie ostrzeżenia, gdyż w tym przypadku użytkownik mógłby oczekiwać podwojonej wartości zmiennej U . W takiej sytuacji należy wywołać bezpośrednio polecenie $S2 = 2 * U$.

W przypadku obliczeń na zmiennych zależnych z użyciem klasy `TwoVarsModel` zmienne oznaczone tym samym symbolem są traktowane jako identyczne.

5.5. Praca z biblioteką PaCAL

Praca z biblioteką **PaCAL** jest stosunkowo prosta, notacja jest zbliżona do tej stosowanej w statystyce. Przeciążone zostały standardowe operatory oraz funkcje, tak aby można było z ich użyciem wykonywać odpowiednie operacje na zmiennych losowych w taki sam sposób, jak zwykle obliczenia na liczbach w standardowych językach programowania. Na przykład wyznacznik macierzy

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

jest równy:

$$a_{11}a_{22} - a_{21}a_{12}.$$

Analogiczny przykład z użyciem biblioteki **PaCAL** znajduje rozkład wyznacznika losowej macierzy o wymiarze 2×2 , której elementy są losowane niezależnie, zgodnie z rozkładem normalnym standaryzowanym.

```

1 from pacal import *
2 A11 = NormalDistr()
3 A12 = NormalDistr()
4 A21 = NormalDistr()
5 A22 = NormalDistr()
6 D = A11 * A22 - A21 * A12
7 D.summary()
8 D.plot()
9 show()
```

```
===== summary =====
```

```
N(0.0,1.0)*N(0.0,1.0)-N(0.0,1.0)*N(0.0,1.0)
```

```
mean = -8.3266726846886741e-17
```

```
var = 1.9999999999997491
```

```
skewness = 0.0
```

```
kurtosis = 5.9999999999829194
```

```
entropy = 1.6931471805599321
```

```
median = 0.0
```

```
mode = 1.2433721467450247e-15
```

```
medianad = 0.6931471805602766
```

```
iqrang(0.025) = 5.99146454710802
```

```
ci(0.05) = (-2.995732273553975, 2.995732273554045)
```

```
range = (-inf, inf)
```



```

tailexp = (-147.68296018663483, -147.68296018663483)
int_err = 9.9920072216264089e-16

```

Jak widać, dokładność obliczeń (`int_err`) jest bardzo wysoka (na poziomie 10^{-15}). Uzyskany wynik można porównać z rezultatem teoretycznym, który w tym przypadku jest rozkładem Laplace'a [113]. Analogiczny wynik można oczywiście uzyskać również dla innych rozkładów, kiedy to zwykle rozkład teoretyczny wyniku nie jest znany. W powyższym przykładzie warto też zwrócić uwagę na fakt, że iloczyn zmiennych o rozkładzie normalnym ma w zerze punkt osobliwy, w którego otoczeniu funkcja gęstości dąży do nieskończoności. Rozkład ten bierze dalej udział w obliczeniach, jak widać bez straty dokładności. Wynikowy rozkład Laplace'a nie ma osobliwości w zerze, natomiast w zerze występuje nieciągłość pochodnej. Dalsze przykłady i zastosowania praktyczne zostaną przedstawione w rozdz. 6.

5.6. Rozkłady wielowymiarowe, obliczenia na zmiennych zależnych

Rozkłady wielowymiarowe dziedziczą po obiekcie `NDdistr`. Dostępne są następujące modele wielowymiarowe:

`NDNormalDistr(mu, Sigma)` – wielowymiarowy rozkład normalny z wektorem średnich `mu` i macierzą kowariancji `Sigma`;

`ClaytonCopula(theta, marginals)` – rozkład zadany kopułą Clayтона z parametrem `theta` $\in (0, \infty)$ oraz z danymi rozkładami brzegowymi `marginals`;

`GumbelCopula(theta, marginals)` – rozkład zadany kopułą Gumbela z parametrem `theta` $\in (1, \infty)$ oraz z danymi rozkładami brzegowymi `marginals`;

`FrankCopula(theta, marginals)` – rozkład zadany kopułą Franka z parametrem `theta` $\in (-\infty, 0) \cup (0, \infty)$ oraz z danymi rozkładami brzegowymi `marginals`;

`IJthOrderStatsNDDistr(X, n, i, j)` – rozkład łączny i -tej i j -tej statystyki pozycyjnej dla próbki o rozmiarze n pobranej w sposób niezależny z populacji o rozkładzie zdefiniowanym zmienną `X`;

`PiCopula(theta, marginals)` – rozkład produktowy.

Obliczenia na zmiennych zależnych wykonujemy, definiując model probabilistyczny (klasa `TwoVarsModel`) składający się z dwuwymiarowego rozkładu łącznego oraz wyrażenia wiążącego zmienne wolne, którego rozkład jest szukany. Ilustruje to poniższy fragment kodu:

```

1 X = UniformDistr(0, 1, sym="X")
2 Y = UniformDistr(0, 1, sym="Y")
3 C = FrankCopula(marginals=[X, Y], theta=-5.0)
4 M = TwoVarsModel(C, X + Y)
5 S = M.eval()

```

```

6 | S.summary()
7 | print "corrcoef=", C.corrcoef(X,Y)

```

```

===== summary =====
PDISTR(0.0,2.0)
      mean = 1.0000000000000031
      var  = 0.059418815324002075
      skewness = -3.784699086628021e-14
      kurtosis = 3.6029647556176303
      entropy = -0.0041618072961233958
      median = 1.0
      mode  = 1.0000000151209214
      medianad = 0.14596861541979195
      iqrang(0.025) = 1.00398153317443
      ci(0.05) = (0.49800923341277636, 1.5019907665872063)
      range = (0.0, 2.0)
      tailexp = (None, None)
      int_err = -2.6645352591003757e-15
corrcoef= -0.643487108056

```

Przykład ten wyznacza sumę dwóch zmiennych, X i Y , opisanych rozkładem zdefiniowanym za pomocą kopuły Franka, z jednostajnymi rozkładami brzegowymi oraz parametrem $\theta = -5.0$ (co, jak widać, odpowiada współczynnikowi korelacji Pearsona równemu -0.6434). Obliczenia są wykonywane metodą `eval` z klasy `TwoVarsModel`. Metoda ta zwraca standardowy obiekt klasy `Distr`.

Metody dostępne dla rozkładów wielowymiarowych zdefiniowanych za pomocą kopuł zebrano w tab. 5.3.

Tab. 5.3. Metody zdefiniowane dla klas dziedziczących po `Copulas`

Metoda	Opis
<code>mode()</code>	wartość modalna, ekstremum funkcji gęstości
<code>cov(i,j)</code>	kowariancja pomiędzy zmiennymi i i j lub (gdy nie podano i, j) macierz kowariancji
<code>corrcoef(i,j)</code>	współczynnik korelacji pomiędzy zmiennymi i i j lub macierz korelacji
<code>tau(i,j)</code>	współczynnik τ Kendalla
<code>rho.s(i,j)</code>	współczynnik ρ Spermana
<code>beta(i,j)</code>	współczynnik β Blomqvista
<code>plot</code>	wykres funkcji gęstości
<code>contour</code>	wykres konturowy funkcji gęstości

6. Przykłady i zastosowania

6.1. Testowanie dokładności

W tej części zostaną przedstawione przykłady numeryczne i oraz możliwe zastosowania praktyczne obliczeń na zmiennych losowych. Wysoka dokładność obliczeń probabilistycznych była jednym z głównych założeń projektowych. Z tego punktu widzenia testowaniu dokładności poświęcono wiele uwagi. Poniżej zostaną przedstawione główne metody testowania biblioteki **PaCAL**, jednak metody te mogą być (i także są) wykorzystywane do testowania również innych bibliotek.

Porównania z rozwiązaniami teoretycznymi

Jednym z najważniejszych punktów odniesienia są formuły teoretyczne. Rozkłady teoretyczne są znane dla wielu szczególnych przypadków. Podobna sytuacja występuje, gdy nie jest znana formuła jawna na postać dystrybucyjną czy funkcji gęstości, ale są znane (lub są proste do wyznaczenia) pewne wartości charakterystyczne zmiennej losowej, jak wartość oczekiwana, momenty wyższych rzędów czy mediana. Duża liczba rozwiązań teoretycznych dotyczących obliczeń arytmetycznych na zmiennych losowych jest opisana w monografii [113], znaczną liczbę przykładów można znaleźć również w różnych podręcznikach z rachunku prawdopodobieństwa, jak na przykład w pracy [28].

Obliczenia probabilistyczne w pakiecie **PaCAL** są wykonywane na funkcjach gęstości. Każdy probabilistyczny rodzaj operacji powinien zachowywać normę $\| \cdot \|_1$ funkcji¹. Ponieważ w pakiecie **PaCAL** nie są stosowane żadne dodatkowe warunki korygujące normę funkcji gęstości w trakcie kolejnych etapów obliczeń, sprawdzenie różnicy

$$E_1 = \left| \int_{-\infty}^{\infty} f(t) dt - 1.0 \right|$$

może być stosowane jako jedna z miar jakości obliczeń numerycznych. Większość praktycznych charakterystyk zmiennej losowej (średnia, przedziały ufności itp.) wyraża się jako całki z funkcji gęstości, dlatego – jak pokazują obliczenia – miara E_1 jest bardzo dobrym wskaźnikiem dokładności dla większości wskaźników i charakterystyk zmiennych losowych.

¹ Jest to tak zwany zerowy moment zmiennej losowej, który zawsze jest równy 1.0.

Do testowania dokładności dogodne są również szczególne typy rozkładów, jak rozkłady stabilne czy rozkłady ilorazowe. W tych przypadkach, podobnie jak przy wykorzystaniu momentu zerowego, obliczenia numeryczne mogą być łatwo sprawdzane za pomocą samych obliczeń numerycznych.

Obliczenia numeryczne z kontrolą błędów

Większość algorytmów metod numerycznych jest zaprojektowana w ten sposób, że umożliwia kontrolę dokładności obliczeń numerycznych w trakcie obliczania. Dotyczy to również algorytmów stosowanych w pakiecie **PaCAL**, zwłaszcza metod całkowania i aproksymacji. Przykładem takiej kontroli dokładności jest wzięcie n -tej sumy częściowej szeregu Czebyszewa. Jako błąd odcięcia bierze się wartość $n + 1$ współczynnika. Tego typu kontrola dokładności nie jest oczywiście w pełni dokładna, jednak – jak pokazuje praktyka – jest wystarczająco dokładna. Zarówno metody całkowania, jak i interpolacji są wyposażone w kontrolę dokładności. Ponadto odpowiednie parametry biblioteki pozwalają obniżyć dokładności obliczeń, co przekłada się na wzrost szybkości. Domyślnie parametry są ustawione na maksymalną dokładność dostępną obliczeniom zmiennopozycyjnym.

Porównania z istniejącymi rozwiązaniami symbolicznymi

W niektórych przypadkach dla oceny dokładności możliwe jest zastosowanie metod wykraczających poza zwykłe obliczenia zmiennopozycyjne. W takich przypadkach mogą być stosowane systemy algebry komputerowej CAS, jak *Mathematica* czy biblioteka **sympy** [117] języka Python oraz obliczenia wielokrotnej precyzji (na przykład biblioteka **mpmath** [56] języka Python oraz również *Mathematica*). W przypadku porównywania dwóch rozwiązań numerycznych, gdy nie jest znane rozwiązanie teoretyczne, punktem odniesienia powinny być obliczenia większej precyzji. Porównanie obliczeń symbolicznych z rozwiązaniami numerycznymi w pakiecie **PaCAL** przedstawiono w podrozdz. 6.2.2. Obliczenia symboliczne mogą być stosowane jedynie w przypadkach, kiedy system CAS jest w stanie znaleźć jawną postać całki.

Histogramy i próbkowanie

W przypadku obliczeń na zmiennych losowych niezależnych łatwo można dostać przybliżone wyniki, wykonując analogiczne obliczenia na próbkach losowych pobranych z odpowiednich rozkładów. Dokładność takich obliczeń zależy od wielkości próbki i zwykle skaluje się z pierwiastkiem kwadratowym z rozmiaru próby. Na przykład próba wielkości 1×10^6 powinna dać dokładność do trzech cyfr znaczących. W części pakietu **PaCAL** dotyczącej arytmetyki zmiennych niezależnych tego typu obliczenia na próbach losowych są wykonywane w tle, dzięki temu są dostępne generatory liczb losowych dla rozkładów złożonych (powstałych w wyniku wykonywania operacji arytmetycznych). Próbkę losową uzyskane w taki sposób otrzymuje się przez wywołanie metody **rand**. Dokładność obliczeń numerycznych zaimplementowanych na zmiennych losowych niezależnych jest o wiele

rzędów wielkości lepsza od dokładności wyników opartych na próbkowaniu. Tego typu kontrola dokładności była wykorzystywana jedynie we wczesnej fazie rozwoju projektu i pozwalała wyeliminować błędy w samej implementacji.

Nieco inaczej jest w przypadku obliczeń na zmiennych zależnych. W takiej sytuacji istnieje konieczność obliczania całek funkcji wielowymiarowych i interpolacji wyników w ogólności wielowymiarowych. Utrzymanie tak wysokiej dokładności, jak to jest w przypadku funkcji jednej zmiennej, jest dużym wyzwaniem i przy rozsądnym czasie obliczeń jest trudne do osiągnięcia. Z drugiej strony standardowym rozwiązaniem w przypadku inferencji probabilistycznej na zmiennych zależnych są metody oparte na próbkowaniu. Odpowiednie schematy próbkowania rozkładów zmiennych zależnych, a zwłaszcza warunkowych nie zostały zaimplementowane w pakiecie **PaCAL**, w tym zakresie do oceny dokładności są wykorzystywane publicznie dostępne pakiety zewnętrzne, jak: **BUGS**, **PyMC** [76,76,95]. Tego typu systemy są praktycznie jedynym punktem odniesienia w przypadku obliczeń na zmiennych zależnych (pomijając modele szczególne oraz bezpośrednie całkowanie symboliczne).

Porównania z istniejącymi rozwiązaniami numerycznymi

Najciekawszą grupę testów stanowią porównania jakości rozwiązań stosowanych w **PaCAL** z rozwiązaniami numerycznymi zaimplementowanymi w różnych statystycznych środowiskach obliczeniowych, jak **SAS**, **R** czy **Python/scipy.stats**. Procedury numeryczne dotyczące zastosowań statystycznych zwykle odnoszą się do szczególnych przypadków i wymagają osobnych implementacji. Zwykle u podstaw wybranej procedury statystycznej leży ustalony rozkład statystyki. W bardziej złożonych wypadkach taki rozkład wymaga wykorzystania przy implementacji znanych rozwinięć asymptotycznych czy przybliżeń w postaci szeregów. Rozwiązania uzyskane za pomocą pakietu **PaCAL** umożliwiają znajdowanie rozkładów takich statystyk w dużej ogólności bez potrzeby znajomości przybliżeń czy rozwinięć. Aproksymacje rozwiązań są obliczane na bieżąco. Mogłoby się wydawać, że rozwiązania zoptymalizowane pod kątem rozwiązania pojedynczego zagadnienia będą działały znacznie lepiej niż ogólne rozwiązania zaimplementowane w pakiecie **PaCAL**. Jednak testy i porównania pokazują, że przyjęte reprezentacje zachowują się pod tym względem bardzo dobrze.

W podrozdziale 6.2.5 przedstawiono na przykład porównanie implementacji rozkładów niecentralnych, które pokazują, że wyniki uzyskane za pomocą pakietu **PaCAL** są bardzo dokładne i mogą być konkurencyjne w stosunku do istniejących metod.

6.2. Przykłady numeryczne

6.2.1. Środowisko testowe

W tej części zostaną zaprezentowane przykłady świadczące o wysokiej jakości zaimplementowanych metod numerycznych. Wszystkie prezentowane przykłady były liczone na maszynie wyposażonej w procesor i7 2.8 GHz, 8 Gb RAM, wersję języka Python v2.7 oraz biblioteki: **numpy** v1.4.1, **matplotlib** v1.0.0, **scipy** v0.9 [61] i **sympy** v0.6.7 [117]. Wszędzie tam, gdzie było to możliwe, punktem odniesienia są powszechnie wykorzystywane środowiska obliczeniowe, jak R v2.13.0 [104], Mathematica v8 [125] czy Matlab v7.11.0.584 (R2010b) (z pakietem **chebfun** v4.1). Przykłady zostały tak dobrane, aby uwypuklić różnice oraz pokazać szeroki zakres możliwości biblioteki **PaCAL**. Należy zaznaczyć, że pewne przykłady obliczeń probabilistycznych można wykonać jedynie za pomocą pakietu **PaCAL** (na przykład obliczenia zagnieżdżone czy obliczenia na zmiennych mieszanych).

6.2.2. Porównanie z pakietami obliczeń symbolicznych

Rozwiązania symbolicznie, takie jak biblioteka **APPL** [39] napisana w Maple czy pakiety statystyczne programu Mathematica (funkcja `TransformedDistribution`, dostępna w wersji 8.x od 15.11.2010 r.), mają tę zaletę, że są w stanie uzyskać dokładne wyniki wtedy, gdy rozkład zmiennej wynikowej można wyrazić jawną formułą. W tym sensie rozwiązania symboliczne stanowią współczesny odpowiednik metod teoretycznych. Zasadnicza trudność polega jednak na tym, że pewne całki nie dają się wyrazić jawnie z użyciem funkcji elementarnych czy nawet funkcji specjalnych. Ilustruje to poniższy fragment sesji w programie Mathematica²:

```
f = TransformedDistribution[x1/x2,{x1 ~ NormalDistribution[0, 1],
                                x2 ~ NormalDistribution[0, 1]}
```

```
CauchyDistribution[0, 1]
```

```
Timing[PDF[f, x]]
```

```
{0., 1/(pi*(1+x^2))}
```

```
g = TransformedDistribution[x1/x2,{x1 ~ NormalDistribution[1, 1],
                                x2 ~ NormalDistribution[1, 1]}
```

```
TransformedDistribution[x1/x2,{x1 ~ NormalDistribution[1, 1], x2 ~ NormalDistribution[1, 1]}
```

```
Timing[PDF[g, x]]
```

```
{64.569, PDF[TransformedDistribution[x1/x2,{x1 ~ NormalDistribution[1, 1],
                                x2 ~ NormalDistribution[1, 1]}, x]}
```

² Prezentowany zapis sesji ma charakter poglądowy; bezpośrednie kopiowanie kodu do programu Mathematica napotka pewne problemy; szczegóły składni można znaleźć w dokumentacji.

W przykładzie tym rozważamy iloraz niezależnych zmiennych o rozkładzie normalnym. Jeżeli zmienne są standaryzowane, to iloraz ma rozkład Cauchy'ego, który jak widać w pierwszym przykładzie, został prawidłowo wyznaczony (zmienna f). Jednak jeżeli zmienne nie są wycelowane (w tym przypadku oba argumenty mają rozkład $N(1, 1)$ – zmienna g), to całkowanie symboliczne napotyka trudności – po 64s wynik nie został znaleziony. W dalszej części zostanie pokazane, że tego typu obliczenia w pakiecie **PaCAL** wykonuje się z łatwością.

Jednak nawet w sytuacji, gdy jawne przedstawienie funkcji istnieje, okazuje się, że korzystanie z niego może powodować znaczne trudności natury numerycznej. Rozważmy stosunkowo prosty teoretycznie przykład obliczania sumy n niezależnych zmiennych losowych o rozkładzie jednostajnym $U(0, 1)$. Obliczenie w pakiecie **PaCAL** sumy $n = 50$ zmiennych niezależnych o rozkładzie jednostajnym $U(0, 1)$ zajmuje około 10s.

```

1 n=50
2 Sn = UniformDistr()
3 for i in range(n-1):
4     Sn += UniformDistr()
5 Sn.summary(show_moments=True)

```

```

mean = 25.0000000000000021
var = 4.16666666666667913
skewness = -3.9341833250541192e-14
kurtosis = 2.9760000000010596
entropy = 2.1324844137323025
median = 25.0
medianad = 1.3803177915547182
iqrang(0.025) = 7.994750349652453
ci(0.05) = (21.002624825173744, 28.997375174826196)
range = (0.0, 50.0)
int_err = -1.5543122344752192e-15

moments:
0 = 1.0000000000000016
1 = 25.0000000000000021
2 = 629.16666666666731
3 = 15937.500000000016
4 = 406301.66666666727
5 = 10423125.000000022
6 = 269040121.72619128
7 = 6986448385.4167023
8 = 182502290574.44583
9 = 4795195848000.0557
10 = 126714325845106.67

```

Rozwiązanie to jest numerycznie dokładne, estymowana dokładność z użyciem normy $\|\cdot\|_1$ wynosi 1.5×10^{-15} , wszystkie wskaźniki mają 13–15 dokładnych cyfr znaczących.

Rozwiązanie teoretyczne (rozkład sumy) w tym przypadku jest funkcją kawałkami wielomianową [113, zadanie 3.12]. *Mathematica* jest w stanie znaleźć w tym przypadku

rozwiązanie teoretyczne, licząc odpowiednie całki w sposób symboliczny. Czas wyznaczenia takiego rozwiązania jest znacznie dłuższy niż w przypadku obliczeń numerycznych – wynosi 10s dla $n = 8$ oraz 180s dla $n = 16$ (jest to czas obliczenia spłotów w sposób symboliczny). Dla porównania biblioteka **PaCAL** dodaje 16 składników w czasie 2.5s. Ponadto forma uzyskanego rozwiązania symbolicznego (zapisanego w bazie jednomianów) nie nadaje się praktycznie do obliczeń numerycznych [51, 113]. Kod źródłowy procedury teoretycznej w języku Python jest dostępny na stronie <http://pacal.sf.net>. O ile w przypadku obliczeń wielokrotnej precyzji **Mathematica** jest w stanie utrzymać wysoką dokładność obliczeń, o tyle w przypadku zwykłych obliczeń zmiennopozycyjnych podwójnej precyzji (wymuszając obliczenia w typie `Real`) dostajemy dla $n = 16$ błąd w normie $\|\cdot\|_1$ na poziomie 1×10^{-6} . Dla większych wartości n teoretyczna formuła staje się niestabilna, osiągając dla $n = 50$ błąd rzędu 1×10^{10} . W przypadku pakietu **PaCAL** obliczenia można z powodzeniem kontynuować dla $n = 100, 200$ i więcej. Oczywiście wraz ze wzrostem n rośnie liczba segmentów i czas obliczeń, jednak wysoka dokładność jest utrzymywana [51].

6.2.3. Porównanie z pakietem **chebfun**

Obecnie nie istnieją publicznie dostępne biblioteki numeryczne do obliczeń na zmiennych losowych oparte na numerycznie dokładnych reprezentacjach zmiennych losowych. Jedyną w zasadzie ogólną biblioteką numeryczną, która pozwala na pewne porównania, jest pakiet **chebfun** napisany w programie **Matlab**. Projekt ten jednak jest pomyślany jako narzędzie do rozwiązywania problemów numerycznych, takich jak równania różniczkowe czy całkowe, i raczej nie był pomyślany jako narzędzie do obliczeń probabilistycznych. Możliwe jest jedynie obliczanie całek skumulowanych (komenda `cumsum`) i spłotów funkcji (komenda `conv`).

Jako pierwszy rozważmy przykład z poprzedniej sekcji – sumę rozkładów jednostajnych. W pakiecie **chebfun** przykładowy kod wygląda następująco:

```
1 u =chebfun(1,[0,1])
2 s = u
3 tic
4 for i = 1:50
5     s = conv(s,u);
6 end
7 err = abs(sum(s) - 1)
8 time = toc
```

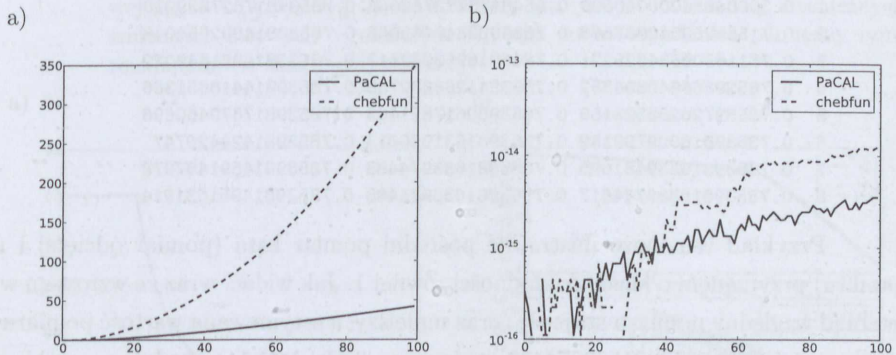
err =

3.1086e-015

time =

86.3410

Jak widać, dokładność obliczeń jest równie wysoka jak w pakiecie **PaCAL**, jednak czas obliczeń jest kilkukrotnie dłuższy. Dokładniejsze porównanie bibliotek **PaCAL** i **chebfun** przedstawiono na rys. 6.1. Pokazano tam błędy w normie $\| \cdot \|_1$ oraz czasy obliczeń w trakcie dodawania kolejnych 100 składników. Ponieważ biblioteki te opierają się na podobnej zasadzie, dokładność obliczeń jest zbliżona. Nieznacznie gorsza dokładność w pakiecie **chebfun** jest spowodowana ustawioną większą tolerancją obliczeń ($100 \cdot \text{eps}(1)$). Trudniej wytłumaczyć dłuższy czas działania. Jedyna zasadnicza różnica wynikająca z porównania kodu źródłowego polega na każdorazowym przycinaniu interpolatora w pakiecie **chebfun** (z użyciem transformacji Czebyszewa opisaną w podrozdz. 4.3.1). Krok ten w bibliotece **PaCAL** jest domyślnie pominięty, jednak są dostępne funkcje umożliwiające takie przycinanie (tab. 4.5).



Rys. 6.1. Porównanie pakietów **PaCAL** i **chebfun** dla sumy niezależnych zmiennych losowych o rozkładzie jednostajnym: a) czas obliczeń w sekundach, b) błąd w normie $\| \cdot \|_1$

Dzięki skupieniu się na aspektach probabilistycznych **PaCAL** znacznie lepiej radzi sobie z osobliwościami oraz rozkładami z ciężkimi ogonami. Na przykład z użyciem pakietu **PaCAL** można wyznaczyć sumę 50 rozkładów Cauchy'ego z utrzymaniem wysokiej precyzji wyniku (13 cyfr znaczących) [51], natomiast pakiet **chebfun** nie radzi sobie nawet z obliczeniem pojedynczego spłotu, nie mówiąc już o rozkładach ze znacznie cięższymi ogonami. Podobnie wygląda sytuacja przy interpolacji funkcji mających bieguny.

6.2.4. Rozwiązania teoretyczne

Iloraz zmiennych normalnych

Wiadomo [28], że iloraz niezależnych zmiennych losowych o wspólnym standaryzowanym rozkładzie normalnym ma wycentrowany rozkład Cauchy'ego. Natomiast arcus tangens zmiennej o rozkładzie Cauchy'ego ma rozkład jednostajny na przedziale $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Poniższy przykład pokazuje obliczenia w pakiecie **PaCAL**. Przypadek $\mu = 0$ odpowiada wycentrowanemu rozkładowi, a $\mu = 1, 2, \dots, 8$ odpowiadają kolejnym przypadkom coraz bardziej niecentralnym.

```

1 for i in range(9):
2     Uc = atan(NormalDistr(mu=i) / NormalDistr(mu=i))
3     print i, repr(Uc.mean()), repr(Uc.median()), repr(Uc.mode())

```

	mu mean	median	mode
0	1.6653345369377e-16	0.0	-1.0957032397306492
1	0.36604655000040509	0.5545473773762666	0.78539817527339228
2	0.71555250540967563	0.7656932340466006	0.78539814092654048
3	0.78116305834625221	0.7846016953032517	0.78539814981549072
4	0.78529866840804352	0.7853841298882738	0.78539814418651366
5	0.78539726285523159	0.7853980617821403	0.78539817679450696
6	0.78539816029799159	0.7853981631060013	0.7853981424426747
7	0.78539816339481605	0.7853981633974483	0.78539814591487978
8	0.78539816339744617	0.7853981633974483	0.78539814951631914

Przykład ten może ilustrować pośredni pomiar kąta (pomiar odciętej i rzędnej punktu) przyrządem o klasie dokładności równej 1. Jak widać, wraz ze wzrostem wartości μ błąd względny pomiaru staje się coraz mniejszy, a estymowana wartość pomiaru zbiega do wartości $\frac{\pi}{4}$ (0.7853981633974483 zapisane w typie double) zgodnie z oczekiwaniami, ponieważ rzędna jest równa odciętej. Widać, że również moda rozkładu w tym przypadku najlepiej odtwarza mierzoną wartość – niezależnie od przesunięcia μ maksimum wynikowego rozkładu jest w punkcie $\frac{\pi}{4}$.

Rozkłady ilorazowe

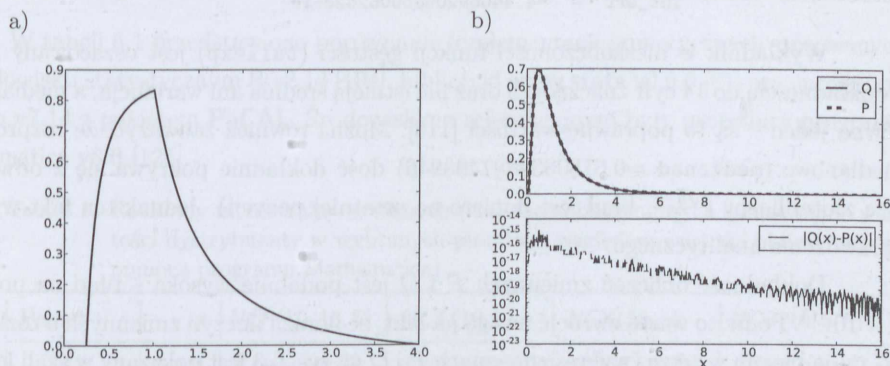
Rozkłady ilorazowe (ang. *ratio distribution*) są wygodnym narzędziem testowania procedur mnożenia i dzielenia zmiennych losowych niezależnych. Niech $X \sim Y$ oraz określmy iloraz zmiennych $R = X/Y$. Jest tu wykorzystywana własność rozkładu zmiennej ilorazowej R : jeżeli $U \sim R$ i $V \sim R$ oraz zmienne U, V są niezależne, to zmienne $P = UV$ oraz $Q = U/V$ mają taki sam rozkład [113].

Dla przykładu rozważmy ilorazy zmiennych o rozkładzie jednostajnym:

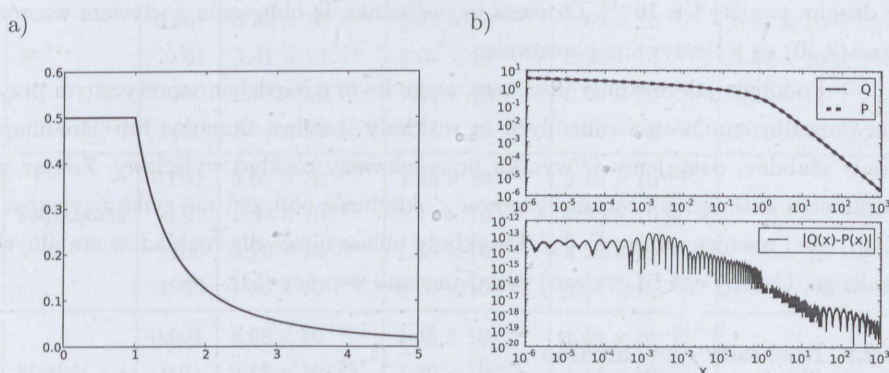
1. $X_1 \sim Y_1 \sim U(0.5, 2)$,
2. $X_2 \sim Y_2 \sim U(0, 1)$.

W pierwszym przypadku rozkład zmiennej ilorazowej $R_1 = X_1/Y_1$ ma nośnik skończony – przedział $[0.25, 4]$. W drugim przypadku zmienna $R_2 = X_2/Y_2$ ma nośnik nieograniczony

(przedział $[0, \infty)$). W obu przypadkach wyznaczono iloczyn (P) i iloraz (Q) zmiennych losowych niezależnych o danych rozkładach ilorazowych. Wyniki przedstawiono na rys. 6.2 i 6.3.



Rys. 6.2. Iloczyn i iloraz rozkładów ilorazowych: a) funkcja gęstości zmiennej ilorazowej $R_1 = U(\frac{1}{2}, 2)/U(\frac{1}{2}, 2)$, b) rozkłady iloczynu P i ilorazu Q niezależnych zmiennych losowych o rozkładzie R_1 oraz punkтова różnica pomiędzy tymi rozkładami (w skali logarytmicznej)



Rys. 6.3. Iloczyn i iloraz rozkładów ilorazowych: a) funkcja gęstości zmiennej ilorazowej $R_2 = U(0,1)/U(0,1)$, b) rozkłady iloczynu P i ilorazu Q niezależnych zmiennych losowych o rozkładzie R_1 oraz punkтова różnica pomiędzy tymi rozkładami (w skali logarytmicznej)

Zauważmy, że rozkład ilorazowy zmiennej R_2 ma ciężki ogon:

```
1 R2 = UniformDistr(0.0,1)/UniformDistr(0.0,1)
2 R2.summary()
```

```
U(0.0,1)/U(0.0,1)
      mean = nan
      var  = nan
      median = 1.0
      medianad = 0.6180339887498946
```



```

iqrangle(0.025) = 19.9500000000001324
ci(0.05) = (0.04999999999999998, 20.0000000000001176)
range = (0.0, inf)
tailexp = (None, -1.999999999999254)
int_err = -4.4408920985006262e-16

```

Wykładnik w nieskończoności funkcji gęstości (`tailexp`) jest oszacowany na -2 z dokładnością do 14 cyfr znaczących oraz nie istnieją średnia ani wariancja, a mediana jest równa jeden – są to poprawne wartości [113]. Można również zauważyć, że rozproszenie medianowe (`medianad` = 0.6180339887498946) dość dokładnie pokrywa się z odwrotnością złotej liczby ($\frac{\sqrt{5}-1}{2}$, błąd jest dopiero na ostatniej pozycji). Jednak ten fakt wymaga sprawdzenia analitycznego.

Dokładność obliczeń zmiennych P i Q jest podobnie wysoka – błąd na poziomie 1×10^{-14} . Ponadto warto zwrócić uwagę na fakt, że iloraz i iloczyn zmiennych o rozkładzie R_2 mają biegun w zerze (wykres zmiennych P i Q na rys. 6.3 jest pokazany w skali logarytmicznej). Ponadto na rys. 6.2b i 6.3b przedstawiono punktową różnicę pomiędzy funkcjami gęstości rozkładów P i Q . Teoretycznie różnica ta powinna być równa zeru. Z wykresu można odczytać, że w pierwszym przypadku różnica ta jest na poziomie 1×10^{-15} , a w drugim poniżej 1×10^{-13} . Oznacza to pośrednio, że obliczenia z użyciem wzorów (2.8) oraz (2.10) są wykonywane poprawnie.

Podobnie jak rozkłady ilorazowe, wygodnym narzędziem teoretycznym przy badaniu procedur sumowania zmiennych są rozkłady stabilne. Sumując lub uśredniając rozkłady stabilne, dostajemy w wyniku przeskalowany rozkład wyjściowy. Znając wartość parametru skalującego, można szacować dokładność obliczeń nie tylko używając normy $\|\cdot\|_1$, ale i również normy $\|\cdot\|_\infty$. Przykłady obliczeniowe dla rozkładów stabilnych (normalnego, Cauchy’ego i Levy’ego) przedstawiono w pracy [51].

6.2.5. Rozkłady niecentralne

Rozkłady niecentralne mają pewne znaczenie w zastosowaniach [113, podrozdz. 9.2]. W bibliotece **PaCAL** oblicza się je bardzo łatwo. Na przykład niecentralny rozkład T można wyznaczyć, stosując poniższy fragment kodu³:

```

1 Num = NormalDistr(mu, 1)
2 Den = (ChiSquareDistr(df) / df) ** 0.5
3 Tnc = Num / Den

```

gdzie μ jest parametrem niecentralnym, a df jest liczbą stopni swobody. Dla $\mu=0$ dostajemy zwykły rozkład t -Studenta. Niecentralny rozkład chi-kwadrat wyznacza się podobnie:

³ Rozkłady niecentralne zdefiniowane w ten sposób zebrano w module **pa-cal.stats.noncentral_distr** (podrozdz. 5.3.2).

```

1 S1 = NormalDistr(sqrt(lmbda), 1)**2
2 S2 = ChiSquareDistr(df - 1)
3 Xnc = S1 + S2

```

W tabeli 6.1 przedstawiono porównanie implementacji numerycznych stosowanych w środowisku statystycznym R v2.13 [104], biblioteki **scipy.stats** v0.9.0 [61] oraz programu Matlab v7.14 z pakietem **PaCAL**. Środowiskiem referencyjnym były procedury programu Mathematica v8.0 [125].

Tab. 6.1. Rozkłady niecentralne w różnych pakietach obliczeniowych (porównanie wartości dystrybuanty w wybranych punktach względem wartości obliczonych za pomocą programu Mathematica)

Pakiet	x	$NCF(10, 15, 5)$	$NCT(10, 5)$	$NCX(10, 5)$	$NCB(10, 5, 5)^{***}$
PaCAL	0.001	1.57×10^{-22}	$2.73 \times 10^{-17*}$	4.71×10^{-26}	6.55×10^{-19}
	0.01	$2.33 \times 10^{-23*}$	$3.08 \times 10^{-17*}$	4.10×10^{-24}	$5.27 \times 10^{-17*}$
	100	$1.11 \times 10^{-16*}$	$1.11 \times 10^{-15*}$	$4.44 \times 10^{-16*}$	$1.11 \times 10^{-15*}$
	1000	$1.11 \times 10^{-16*}$	$1.11 \times 10^{-15*}$	$4.44 \times 10^{-16*}$	$9.99 \times 10^{-16*}$
R **	0.001	3.65×10^{-20}	$2.72 \times 10^{-17*}$	$9.63 \times 10^{-35*}$	$6.61 \times 10^{-21*}$
	0.01	3.41×10^{-13}	$3.04 \times 10^{-17*}$	$2.76 \times 10^{-30*}$	3.54×10^{-14}
	100	2.15×10^{-10}	7.44×10^{-13}	$0.00 \times 10^{-16*}$	1.69×10^{-10}
	1000	3.14×10^{-10}	7.45×10^{-11}	$0.00 \times 10^{-16*}$	2.10×10^{-10}
scipy.stats	0.001	$2.63 \times 10^{-26*}$	1.45×10^{-9}	2.13×10^{-20}	—
	0.01	2.42×10^{-17}	1.49×10^{-08}	7.30×10^{-28}	—
	100	1.26×10^{-05}	1.00×10^{-11}	4.20×10^{-07}	—
	1000	1.26×10^{-05}	$0.00 \times 10^{-16*}$	4.20×10^{-07}	—
Matlab	0.001	$8.68 \times 10^{-30*}$	1.45×10^{-09}	$1.63 \times 10^{-34*}$	—
	0.01	$2.17 \times 10^{-24*}$	1.49×10^{-08}	$4.34 \times 10^{-30*}$	—
	100	4.69×10^{-14}	$4.22 \times 10^{-15*}$	$0.00 \times 10^{-16*}$	—
	1000	4.64×10^{-14}	$0.00 \times 10^{-16*}$	$0.00 \times 10^{-16*}$	—

* Najwyższa dokładność lub pełna dokładność maszynowa.

** W niektórych przypadkach dostajemy ostrzeżenie o możliwej utracie dokładności.

*** Dla niecentralnego rozkładu beta porównano wartości w punktach: 0.01, 0.1, 0.9, 0.95.

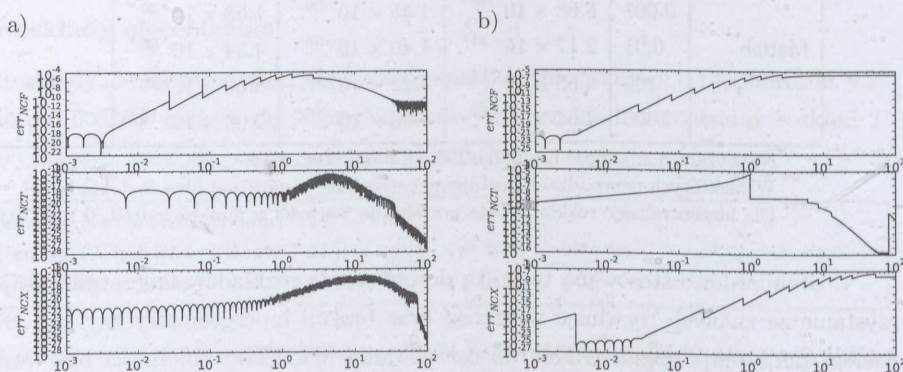
Standardowo stosowaną techniką do obliczania rozkładów niecentralnych jest korzystanie ze znanych rozwinięć w szeregi oraz funkcji hipergeometrycznych [1, 87, 113]. Szeregi takie mają często trudne do oszacowania przedziały zbieżności oraz reszty, co powoduje, że są trudne w implementacji. Program Mathematica również standardowo posługuje się szeregami hipergeometrycznymi przy wykonywaniu bardziej złożonych operacji związanych z całkowaniem, które nie wyrażają się w skończonej postaci za pomocą znanych funkcji elementarnych. Jednak dodatkowo Mathematica posługuje się liczbami

zapisanymi z użyciem wielokrotnej precyzji, dzięki czemu jest w stanie uzyskać wyniki na poziomie dokładności nieosiągalnej dla zwykłych obliczeń zmiennopozycyjnych (w typie `double`)).

Biblioteka **PaCAL** wykorzystuje odmienne podejście numeryczne. W pakiecie tym całki z parametrem definiujące odpowiednie operacje na zmiennych losowych są liczone bezpośrednio, a następnie – aproksymowane wielomianami Czebyszewa. Wyniki uzyskiwane w ten sposób są porównywalne do wyników programu **Mathematica** w zakresie precyzji obliczeń zmiennopozycyjnych; pewną utratę dokładności względnej można zaobserwować dopiero dla wartości funkcji bliskich zera.

W tabeli 6.1 przedstawiono różnice pomiędzy wartością dystrybucyjną w zadanych punktach, obliczoną za pomocą znanych pakietów statystycznych (**R**, **Matlab** oraz **scipy.stats**), a referencyjną wartością obliczoną z użyciem programu **Mathematica**. Jak widać, jakość obliczeń pakietu **PaCAL** jest znacznie wyższa niż biblioteki **scipy.stats** oraz wyższa niż pakietu **R**. W programie **Matlab** widać utratę dokładności dla niecentralnego rozkładu t . Dokładność bezwzględna biblioteki **PaCAL** we wszystkich analizowanych przypadkach była mniejsza niż 1×10^{-14} , dokładność względna, zwłaszcza dla bardzo małych wartości funkcji (w otoczeniu zera), była nieco wyższa.

Na rysunku 6.4 przedstawiono punktowe różnice pomiędzy funkcjami gęstości i dystrybucyjnymi wyznaczanymi na podstawie pakietów **scipy.stats** oraz **PaCAL**. Jeżeli chodzi o funkcje gęstości, to jakość obliczeń dla rozkładów niecentralnych T (NCT) i chi-kwadrat (NCX) w obu przypadkach jest porównywalnie wysoka i bliska precyzji maszynowej. Dla rozkładu niecentralnego F (NCF) obliczenia z użyciem **scipy.stats** dają jedynie 5 – 6 cyfr znaczących dokładności. Ponadto na rys. 6.4 wyraźnie widać granice przedziałów aproksymacji. W przypadku aproksymacji dystrybucyjnej w pakiecie **scipy.stats** w zasadzie



Rys. 6.4. Błędy przy obliczaniu rozkładów niecentralnych (chi-kwadrat, t , F) w pakiecie **scipy.stats** dla: a) funkcji gęstości, b) dystrybucyjnej; dla dystrybucyjnej jest obserwowana wyraźna utrata jakości – błąd na poziomie 1×10^{-6} .

wszystkie funkcje są źle aproksymowane. Uzyskana dokładność jest na poziomie 1×10^{-6} . Na przykładzie tym widać, jak trudne w implementacji są funkcje hipergeometryczne, na których opierają się standardowe implementacje rozkładów niecentralnych. Jakościowo nieco lepsze implementacje, jak widać, można znaleźć w pakietach R i Matlab, chociaż i w tych przypadkach widać pewną utratę dokładności, na przykład przy niecentralnym rozkładzie t .

W dalszej części zostaną zaprezentowane możliwe zastosowania biblioteki w problemach praktycznych.

6.3. Możliwości i zastosowania pakietu PaCAL

6.3.1. Wnioskowanie statystyczne

Za pomocą pakietu **PaCAL** można łatwo wyznaczyć rozkład dowolnej statystyki dla próby losowej pobranej w sposób niezależny, która jest wynikiem operacji arytmetycznych na elementach próby, jeżeli założony zostanie rozkład pojedynczej obserwacji. Podejście takie jest konkurencyjne, zwłaszcza w przypadkach, w których nie są znane teoretyczne rozkłady statystyk. Standardowe rozwiązania polegają na wykorzystaniu aproksymacji różnego rodzaju. Aproksymacje motywowane probabilistycznie, jak na przykład zbieżność do rozkładu normalnego, są poprawne zwykle dla dużych prób. Dokładniejsze aproksymacje numeryczne wymagają znajomości pewnych faktów teoretycznych dotyczących szukanego rozkładu, jak na przykład znajomość szeregu oraz oszacowanie błędu odcięcia.

W pracach [51, 68] przedstawiono zastosowanie pakietu do wyznaczania dokładnego rozkładu statystyki Hilla [18] szacującej wykładnik w nieskończoności dla rozkładów opisanych asymptotycznie rozkładem Pareto. Prezentowane rozwiązanie pracuje stabilnie nawet dla rozkładów o wykładnikach z przedziału pomiędzy 1 a 2. Rozkłady takie mają ciężkie ogony i są w ogólności trudne zarówno do aproksymacji, jak i do całkowania. Poniżej zostanie przedstawione zastosowanie rozkładów niecentralnych do wyznaczania mocy testu.

Wyznaczanie mocy testu t -Studenta

Rozkłady niecentralne prezentowane w poprzednim podrozdziale mogą być wykorzystywane do wyznaczania mocy testów [93]. Załóżmy, że mamy próbkę $\{x_1, \dots, x_n\}$ pobraną z populacji o rozkładzie normalnym o średniej μ . Hipoteza zerowa, będąca podstawą budowy statystyki testowej, mówi, że $H_0: \mu = \mu_0$. Rozważmy test dwustronny z hipotezą alternatywną $H_1: \mu \neq \mu_0$. W przypadku prawdziwości hipotezy H_0 statystyka $\frac{\bar{X} - \mu}{S/\sqrt{n}}$ ma rozkład t -Studenta z $n-1$ stopniami swobody ($\bar{X}$ i S oznaczają średnią i odchylenie standardowe z próby). Odrzucamy H_0 , jeżeli wartość statystyki testowej na próbce wpada do obszaru krytycznego $(-\infty, t_{\alpha, n-1}) \cup (t_{\alpha, n-1}, \infty)$, gdzie α jest założonym poziomem

istotności. Może zdarzyć się, że odrzucamy hipotezę zerową w przypadku, gdy jest ona prawdziwa. Jednak w takiej sytuacji znamy dokładnie prawdopodobieństwo tego błędu – wynosi ono α (jest to błąd pierwszego rodzaju) [67].

Rozkład statystyki testowej został wyznaczony przy założeniu $\mu = \mu_0$. Jeżeli średnia w populacji jest równa μ_1 zamiast μ_0 , to wtedy statystyka testowa ma rozkład niecentralny t z parametrem niecentralności równym $\delta = \frac{\mu_1 - \mu_0}{S/\sqrt{n}}$ [93]. Dla przykładu przyjmijmy, że $\mu_1 - \mu_0 = 1.5$, $n = 20$, $S = 2$.

```
1 delta = 1.5; sd = 2.0; n = 20.0; ncp = delta / (sd / sqrt(n))
2 t = StudentTDistr(df=n - 1)
3 nct = NoncentralTDistr(df=n - 1, mu=ncp)
4 t_alpha = t.quantile(0.975)
5 print "      ncp =", ncp
6 print "  t_alpha=", t_alpha
7 print "      moc=", 1 - (nct.cdf(t_alpha) - nct.cdf(-t_alpha))
8 print "błąd II-rodz.", (nct.cdf(t_alpha) - nct.cdf(-t_alpha))
```

```
      ncp = 3.35410196625
  t_alpha = 2.09302405441
      moc = 0.8888478174
błąd II-rodzaju = 0.1111521826
```

Analogiczne obliczenia w pakiecie R wyglądają następująco:

```
1 power.t.test(n=20,delta=1.5,sd=2,strict=TRUE,type="one.sample")
```

One-sample t test power calculation

```
      n = 20
  delta = 1.5
      sd = 2
sig.level = 0.05
  power = 0.8888478
alternative = two.sided
```

Dokładność obliczeń w obu przypadkach jest wysoka i porównywalna – zgodność do 14 cyfr znaczących. Modyfikując nieznacznie kod, możemy wyznaczyć taki rozmiar próby lub graniczną wartość różnicy $\mu_1 - \mu_0$, aby moc osiągnęła żadaną wartość. W tym celu należy rozwiązać numerycznie odpowiednie równania. Analogicznie można wyznaczać moc testu χ^2 .

6.3.2. Pomiary wielkości fizycznych

W problemach technicznych często dokonuje się pomiaru wielkości fizycznych, na których są następnie wykonywane obliczenia, na przykład w przypadku pomiarów średnich. Pomiary takie są jednak obciążone błędami o charakterze losowym (szumem),

co powoduje, że wyniki obliczeń są zmiennymi losowymi. Tradycyjnie szacuje się średnie i odchylenie standardowe wyników za pomocą metody delta. W przypadku wielkości będących ilorazami mierzonych wielkości takie rozwiązanie jest poprawne, jeżeli zmienność⁴ mierzonych wielkości jest mała. W przypadku pomiarów małych wielkości, gdzie wartość wielkości mierzonej jest bliska poziomowi błędów pomiarowych, wyznaczanie dokładnych rozkładów obliczanych zmiennych może mieć duże znaczenie praktyczne.

Rozważmy przykład⁵, w którym interesuje nas pomiar pośredni mocy wydzielonej na rezystorze z uwzględnieniem zmian rezystancji pod wpływem temperatury. Moc P dana jest wzorem:

$$P = \frac{U^2}{R_0(1 + \alpha(T - T_0))},$$

gdzie: U (V) jest napięciem mierzonym na stykach rezystora, T (°C) – temperaturą w chwili pomiaru, T_0 (°C) – temperaturą początkową, R_0 (Ω) – rezystancją w temperaturze początkowej oraz α ($\frac{1}{^\circ\text{C}}$) – współczynnikiem temperaturowym rezystora. Przyjmijmy wartości pomiarów: $U = 10 \pm 1$ V, $T = 30 \pm 0.5^\circ\text{C}$, $T_0 = 20 \pm 0.5^\circ\text{C}$, $R_0 = 47 \pm 4$ Ω oraz znajomość (z pewną dokładnością) wartości współczynnika termicznego $\alpha = 0.003 \pm 0.002 \frac{1}{^\circ\text{C}}$.

Za pomocą pakietu **PaCAL** z łatwością można znaleźć numerycznie dokładny rozkład wielkości P przy założonych modelach niepewności pomiarów zmiennych wejściowych (U , T , T_0 , α , R_0). Poniżej przedstawiono przykładowy kod definiujący zmienne oraz wykonujący obliczenia.

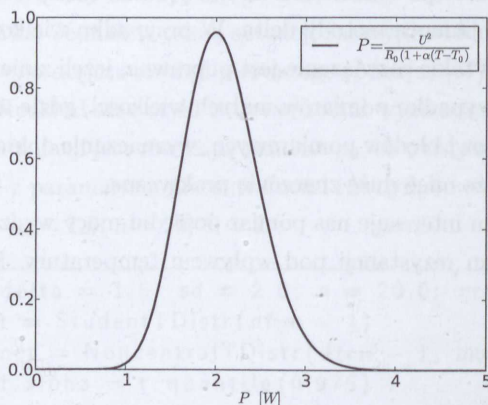
```

1 U = NormalDistr(10, 1)
2 T = UniformDistr(29.5, 30.5)
3 T0 = UniformDistr(19.5, 20.5)
4 alpha = UniformDistr(0.001, 0.005)
5 R0 = UniformDistr(43, 51)
6 P = U ** 2 / (R0 * (1 + alpha * (T - T0)))
7 P.summary()
```

Dokładność obliczeń jest na poziomie 1×10^{-12} . W wyniku dostajemy pełen rozkład, na podstawie którego można wyznaczyć interesujące wskaźniki liczbowe: $P = 2.068 \pm 0.334$ W oraz 95-procentowy przedział ufności wyniku, który wynosi (1.311, 2.958). Uzyskany rozkład zmiennej P przedstawiono na rys. 6.5.

⁴ Zmienność zmiennej losowej (współczynnik zmienności) definiuje się jako iloraz odchylenia standardowego przez średnią.

⁵ Przykład ten wzięto z pracy [59, podrozdz. 4.1.1]; sugerowane w tym dokumencie metody znajdowania niepewności wynikowej polegają na próbkowaniu lub na zastosowaniu metody delta [59, rozdz. 5].



Rys. 6.5. Rozkład mocy P , mierzonej w sposób pośredni, w zależności od bezpośrednich pomiarów zmiennych U , T , T_0 , α , R_0

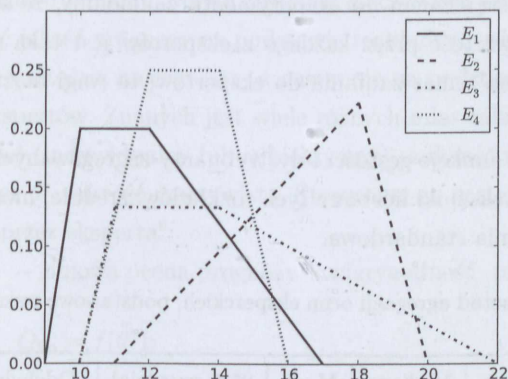
6.3.3. Agregacja ocen eksperckich

Wiele zmiennych ekonomii jest mierzonych przez ekspertów, którzy szacują prognozy dotyczące ustalonego wskaźnika w pewnym okresie. Jeżeli założymy, że wielkość niepewna szacowana przez eksperta może być modelowana w sposób probabilistyczny, to oszacowanie możemy traktować jako zmienną losową o rozkładzie wskazanym przez eksperta. Dla podniesienia dokładności oszacowania stosuje się często ankietyzację większej liczby ekspertów. Mając wyniki oszacowań uzyskane od kilku ekspertów, należy zastosować wybraną metodę agregacji⁶. Warto zwrócić uwagę, że z technicznego punktu widzenia wielkość wskazana przez eksperta może być traktowana jako wartość pomiaru. Jednak pomiędzy oszacowaniem eksperta a pomiarem przyrządem pomiarowym istnieje zasadnicza różnica. Gdy dokonujemy pomiaru, znamy zwykle klasę dokładności przyrządu. Natomiast oszacowanie wskazane przez eksperta ma charakter indywidualny (subiektywny). W pomiarach zwykle wyżej cenimy wskazanie przyrządem o wyższej klasie dokładności – mającym mniejszą wariancję pomiaru. Natomiast mając dane dwa oszacowania, wskazane przez dwóch różnych ekspertów, różniące się rozproszeniem (szerokością przedziału), nie można jednoznacznie ocenić, które z tych oszacowań jest lepsze⁷. Ocenę prognozy można wykonać dopiero po fakcie – po odkryciu prawdziwej wartości zmiennej prognozowanej. W przypadku ocen eksperckich należy założyć, że eksperci dokonują oszacowań w sposób rzetelny, zgodnie ze swoją najlepszą wiedzą i doświadczeniem. Różnice w ocenach mogą być na przykład związane z błędną interpretacją dostępnych informacji, jak również z ilością dostępnych informacji.

⁶ Niektóre z metod agregacji można znaleźć w pracy [91, rozdz. 9].

⁷ Klasę dokładności przyrządu wyznacza się zwykle w serii eksperymentów w laboratorium. W stosunku do eksperta zwykle nie znamy tego parametru a priori.

Rozważmy sztuczny przykład, w którym czterech ekspertów $E_1, \dots, E_4$ szacuje przedziałowo wielkość stopy bezrobocia w kolejnym roku. Przypuśćmy, że wskazania kolejnych ekspertów dane są kolejno przedziałami: $[9\%, 17\%]$, $[11\%, 20\%]$, $[10\%, 22\%]$, $[10\%, 16\%]$. Załóżmy, że oceny ekspertów w podanych przedziałach mają rozkłady trapezowe, jak na rys. 6.6.



Rys. 6.6. Rozkłady stopy bezrobocia wskazane przez ekspertów

Poniższy fragment kodu definiuje rozkłady wskazane przez ekspertów oraz agreguje je, wykorzystując odpowiednie metody uśredniania (średnie: arytmetyczna (A_1), geometryczna (A_2), harmoniczna (A_3), kwadratowa (A_4)). Przedstawiono dodatkowo miary agregacji działające bezpośrednio na rozkładach: agregację liniową (A_5), która odpowiada probabilistycznej alternatywie lub mieszanke rozkładów (ang. *linear opinion pool*), oraz agregację iloczynową (A_6), która odpowiada koniunkcji wskazań lub mieszanke iloczynowej (ang. *logarithmic opinion pool*) [91, rozdz. 9].

```

1 E1 = TrapezoidalDistr(9, 10, 12, 17)
2 E2 = TrapezoidalDistr(11, 18, 18, 20)
3 E3 = TrapezoidalDistr(10, 11, 14, 22)
4 E4 = TrapezoidalDistr(10, 12, 14, 16)
5
6 A1 = 0.25 * E1 + 0.25 * E2 + 0.25 * E3 + 0.25 * E4
7 A2 = E1 ** 0.25 * E2 ** 0.25 * E3 ** 0.25 * E4 ** 0.25
8 A3 = 4 / (1 / E1 + 1 / E2 + 1 / E3 + 1 / E4)
9 A4 = ((E1 ** 2 + E2 ** 2 + E3 ** 2 + E4 ** 2) / 4) ** 0.5
10 A5 = MixDistr([0.25, 0.25, 0.25, 0.25], [E1, E2, E3, E4])
11 f6 = E1.get_pieewise_pdf()
12 for Ei in [E2, E3, E4]:
13     f6 *= Ei.get_pieewise_pdf()
14 f6 = f6 * (1 / f6.integrate(-Inf, Inf))
15 A6 = FunDistr(fun=f6, breakPoints=f6.getBreaks())

```


Zagregowane zmienne $A_1 - A_4$ można łatwo uzyskać, stosując operacje arytmetyczne. Zmienna A_5 jest mieszkanką rozkładów, a zmienna A_6 nie ma odpowiednika wśród operacji probabilistycznych, jednak można ją uzyskać, wykonując działania (iloczyn i normalizację) bezpośrednio na rozkładach. Warto zwrócić uwagę, że agregacja iloczynowa w przypadku ocen wykluczających się nie jest zdefiniowana poprawnie.

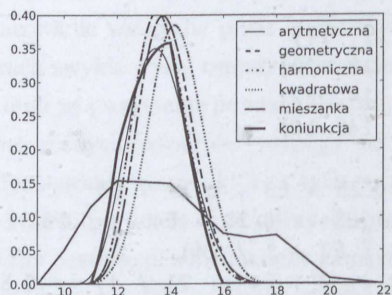
W eksperymencie tym każdemu ekspertowi przypisano a priori jednakową wagę równą $\frac{1}{4}$. Oznacza to, że przed przystąpieniem do eksperymentu zakładamy, że szansa poprawnego oszacowania nieznaney wartości przez każdego z ekspertów jest taka sama. Gdybyśmy mieli dodatkową wiedzę na temat zaufania do ekspertów, to wagi można by odpowiednio zmodyfikować.

Na rysunku 6.7 przedstawiono funkcje gęstości i dystrybuanty zagregowanych rozkładów, a w tab. 6.2 – podstawowe wskaźniki liczbowe tych rozkładów: średnie, mediany, mody, przedział ufności oraz odchylenia standardowe.

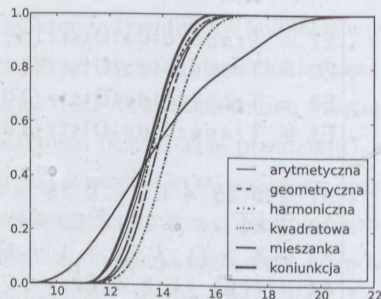
Tab. 6.2. Porównanie różnych metod agregacji ocen eksperckich, podstawowe wskaźniki liczbowe

Metoda agregacji	Średnia	Mediana	Moda	95% przedział ufności	Odchylenie standardowe
A_1 śr. arytmetyczna	13.60	13.55	13.45	(11.87, 15.49)	1.10
A_2 śr. geometryczna	13.84	13.81	13.72	(12.30, 15.51)	0.97
A_3 śr. harmoniczna	13.66	13.61	13.50	(12.15, 15.33)	0.96
A_4 śr. kwadratowa	14.22	14.21	14.15	(12.56, 15.95)	1.02
A_5 mieszanica rozkładów	14.03	13.70	12.00	(10.40, 18.62)	2.54
A_6 koniunkcja rozkładów	13.43	13.45	14.00	(11.81, 15.07)	1.00

a)



b)



Rys. 6.7. Wyniki agregacji ocen eksperckich z użyciem metod: średnia arytmetyczna (A_1), średnia geometryczna (A_2), średnia harmoniczna (A_3), średnia kwadratowa (A_4), agregacja liniowa (A_5), agregacja iloczynowa (A_6); przedstawiono: a) funkcje gęstości, b) dystrybuanty

Jak można zaobserwować, średnia kwadratowa jest przesunięta o ok. 0.5% względem pozostałych średnich. Agregacja iloczynowa (A_6) dość dokładnie pokrywa się ze średnią arytmetyczną, geometryczną i harmoniczną – różnice są na poziomie kilku procent. Największy przedział ufności daje w tym przypadku średnia harmoniczna. Agregacja liniowa (A_5) daje najszerszy przedział i najbardziej odbiegający od pozostałych.

Przed przystąpieniem do eksperymentu każdemu ekspertowi zostało przypisane jednakowe zaufanie. Mając dostępną rzeczywistą wartość prognozowanej wielkości, można ocenić jakość prognozy na podstawie rozkładów wskazanych przez ekspertów w porównaniu z wartością rzeczywistą, a następnie na podstawie tej oceny zmieniać swoje zaufanie do ekspertów. Znanych jest wiele różnych miar takiej oceny prognoz. Oznaczmy przez f rozkład (indywidualny lub subiektywny) szukanego wskaźnika wskazany przez eksperta, a przez θ^* wartość rzeczywistą. Stosowane są następujące miary jakości wyniku wskazanego przez eksperta⁸:

- liniowa ocena prognozy – wiarygodność (ang. *linear score*) :

$$Q_{\text{lin}} = f(\theta^*);$$

- logarytmiczna ocena prognozy – logarytm wiarygodności (ang. *logarithmic score*):

$$Q_{\text{log}} = \log f(\theta^*);$$

- kwadratowa ocena prognozy (ang. *quadratic score*):

$$Q_{\text{sqr}} = 2f(\theta^*) - \int_{-\infty}^{\infty} f^2(\theta) d\theta;$$

- sferyczna ocena prognozy (ang. *spherical score*):

$$Q_{\text{spher}} = \frac{f(\theta^*)}{\sqrt{\int_{-\infty}^{\infty} f^2(\theta) d\theta}};$$

- ocena prognozy z wykorzystaniem stopniowanego prawdopodobieństwa (ang. *ranked probability score*):

$$Q_{\text{rank_prob}} = \int_{-\infty}^{\theta^*} F^2(\theta) d\theta + \int_{\theta^*}^{\infty} (1 - F(\theta))^2 d\theta.$$

Używając pakietu **PaCAL**, każdą z tych miar można obliczyć stosunkowo prosto. Wymagają one wykonania punktowych operacji na funkcjach gęstości i dystrybuantach oraz ewentualnie całkowania. Na przykład miarę $Q_{\text{rank_prob}}$ można wyznaczyć następująco:

⁸ Miary opisano na podstawie pracy [91, rozdz. 8]; można tam znaleźć również szczegółowe informacje źródłowe.


```

1 F2 = Ai.get_pieciwise_cdf() ** 2.0
2 F2_ = (1 - Ai.get_pieciwise_cdf()) ** 2.0
3 Q_rank_prob = F2.integrate(-Inf, theta) + F2_.integrate(theta, Inf)

```

W tabeli 6.3 można prześledzić, jak zmieniałyby się oceny dokładności wskazań ekspertów E_1, E_2, E_3, E_4 , gdyby rzeczywista wartość prognozowanej wielkości (stopy bezrobocia) była, odpowiednio, $\theta^* = 11.0\%, 13.0\%, 15.0\%, 18.0\%$. Wyniki przedstawiono dla różnych miar oceny prognozy opisanych powyżej. Jak widać, oceny prognoz z użyciem różnych miar $Q_{(\cdot)}$ są w dużym stopniu zgodne oraz pokrywają się z rozkładami wskazanymi przez ekspertów, przedstawionymi na rys. 6.6. Jedyna różnica pomiędzy ocenami jest dla przypadku $\theta^* = 15$, gdzie miary Q_{lin} i Q_{log} lepiej oceniają eksperta E_2 , natomiast miary: Q_{sqr} , Q_{spher} , $Q_{\text{rank_prob}}$ wyżej oceniają wskazania eksperta E_3 . Przyglądając się rys. 6.6, trudno jednoznacznie określić, która z miar oceny jest lepsza, a różnice pomiędzy wartościami wskaźników $Q_{(\cdot)}$ dla ekspertów E_2 i E_3 są tak naprawdę zbliżone.

Tab. 6.3. Porównanie różnych metod oceny prognoz eksperckich dla różnych wartości rzeczywistych θ^*

θ^*	Ekspert	Q_{lin}	Q_{log}	Q_{sqr}	Q_{spher}	$Q_{\text{rank_prob}}$
11.00	E_1	0.20	-1.61	0.24	0.50	0.66
	E_2	0.00	$-\infty$	-0.15	0.00	4.24
	E_3	0.13	-2.01	0.16	0.41	2.11
	E_4	0.12	-2.08	0.04	0.27	1.30
13.00	E_1	0.16	-1.83	0.16	0.40	0.65
	E_2	0.06	-2.76	-0.02	0.16	2.32
	E_3	0.13	-2.01	0.16	0.41	0.91
	E_4	0.25	-1.39	0.29	0.55	0.34
15.00	E_1	0.08	-2.53	-0.00	0.20	1.90
	E_2	0.13	-2.06	0.11	0.33	0.91
	E_3	0.12	-2.15	0.13	0.36	0.77
	E_4	0.12	-2.08	0.04	0.27	1.30
18.00	E_1	0.00	$-\infty$	-0.16	0.00	4.79
	E_2	0.22	-1.50	0.30	0.58	0.87
	E_3	0.07	-2.71	0.03	0.20	2.22
	E_4	0.00	$-\infty$	-0.21	0.00	4.26

6.4. Obliczenia na zmiennych zależnych

6.4.1. Rozstęp z próby, rozstęp międzykwantylowy

W kontroli jakości produkcji miarą powtarzalności wyrobu (na przykład pod względem wymiaru lub wagi) może być wariancja estymowana na próbie, jednak znacznie ciekawsze jest oszacowanie, jaka część produkcji mieści się w zakresie od $X_{\min}$ do $X_{\max}$ lub w jakim przedziale mieści się 95% produkcji. W związku z tym rozstęp z próby definiuje się jako różnicę pomiędzy skrajnymi wartościami $X_{(n)} - X_{(1)}$, analogicznie definiuje się różnicę pomiędzy i -tą i j -tą statystyką pozycyjną. Mając zdefiniowany rozkład łączny statystyk pozycyjnych, można, stosując mechanizmy obliczeniowe dla zmiennych zależnych, wyznaczyć rozkład dowolnej operacji na tych zmiennych. Poniższy przykład został zaczerpnięty z instrukcji obsługi pakietu **QC** programu SAS [106], służącego do statystycznej kontroli jakości. Załóżmy, że pobrano w sposób niezależny próbkę wyrobów o rozmiarze $n = 5$ oraz następnie pomierzono odchylenia wyrobów od wartości wzorcowej, a interesuje nas rozkład różnicy pomiędzy wartością maksymalną a minimalną. Zakładamy, że rozkład pojedynczej obserwacji jest normalny, o wartości średniej zero oraz zadany poziomie wariancji 1. Wyznamy w takim modelu rozkłady zmiennych $R_1 = X_{(4)} - X_{(2)}$ oraz $R_2 = X_{(5)} - X_{(1)}$. W pakiecie **PaCAL** zmienną R_2 można obliczyć w następujący sposób:

```

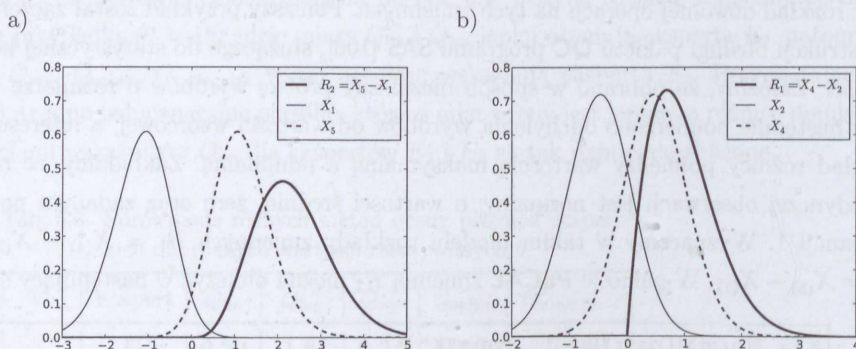
1 X = NormalDistr(0, 1, sym="X")
2 J15 = IJthOrderStatsNDDistr(X, n=5, i=1, j=5)
3 X1, X5 = J15.marginals
4 M2 = TwoVarsModel(J15, X5 - X1)
5 R2 = M2.eval()
6 figure()
7 J15.plot()
8 J15.summary()
9 show()

mean = 2.3259289472810414
var = 0.74663760093428744
skewness = 0.46551383286459874
kurtosis = 3.1691477882127348
median = 2.2568824930262203
mode = 1.9999999697581552
medianad = 0.5844916749951601
iqrang(0.025) = 3.3473546743420313
ci(0.05) = (0.8496716716531508, 4.197026345995182)
range = (-inf, inf)
tailexp = (0, -117.85789031893026)
int_err = -4.4408920985006262e-16

```

W wyniku dostajemy rozkład zmiennej R_2 , dla której wyznaczono podstawowe statystyki podsumowujące. Uzyskane wyniki można porównać z wynikami pakietu SAS/**QC** [106, s. 2237–2238] i są one całkowicie zgodne ($R2.mean()=2.3259289472810414$ oraz

$R2.std()=0.86408194109950442$). Warto zaznaczyć, że funkcje D2 i D3 pakietu **QC** umożliwiają wyznaczenie jedynie średniej i odchylenia standardowego rangi. W pakiecie **PaCAL** znajdujemy pełen rozkład statystyki rangowej oraz możemy wyznaczyć rozkład różnicy dowolnych innych statystyk pozycyjnych (na przykład można znaleźć rozkład dla rozstępu międzykwantylowego z próby). Na rysunku 6.8 przedstawiono rozkład zmiennych pozycyjnych $X_{(2)}, X_{(4)}, X_{(1)}, X_{(5)}$ oraz różnice pomiędzy nimi: $R_1 = X_{(4)} - X_{(2)}$, $R_2 = X_{(5)} - X_{(1)}$. Dokładność obliczeń ponownie jest bardzo wysoka, bliska precyzji maszynowej. Prezentowany przykład dotyczy szczególnego aspektu kontroli jakości, a szczegółowy przegląd procedur można znaleźć w pracach [45, 118].



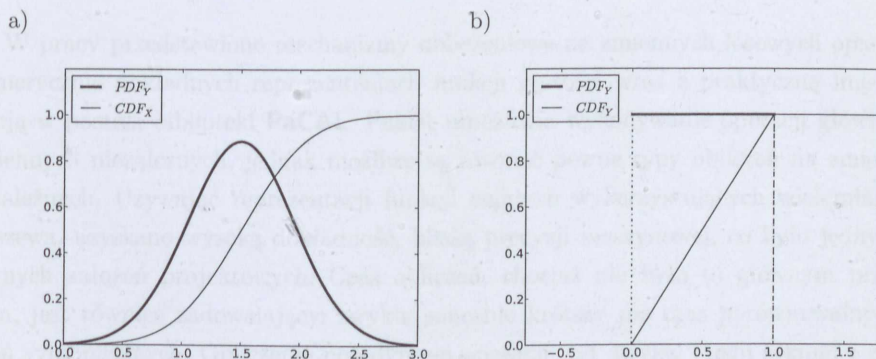
Rys. 6.8. Rozkład zmiennych pozycyjnych: a) $X_{(1)}, X_{(5)}$ oraz różnicy $R_2 = X_{(5)} - X_{(1)}$, b) $X_{(2)}, X_{(4)}$ oraz $R_1 = X_{(4)} - X_{(2)}$

6.4.2. Skrajne możliwe wartości sumy zmiennych losowych

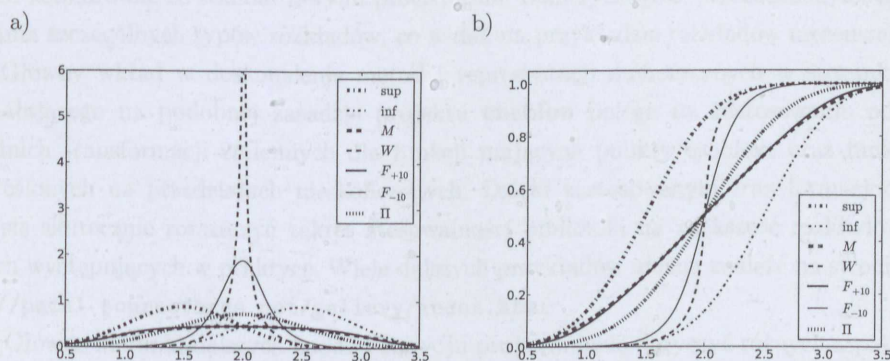
W pewnych sytuacjach nie dysponujemy rozkładem łącznym, natomiast jesteśmy zainteresowani podaniem możliwych ograniczeń na rozkład wyniku lub skrajnych możliwych wartości. W przypadku pewnych operacji można zastosować metody przedstawione w podrozdz. 3.5.3 i 3.5.4.

Rozważmy dwie zmienne losowe: $X \sim U[0, 1]$ i $Y \sim N(1.5, 0.45)$, przedstawione na rys. 6.9. Bez podania dodatkowej informacji o rozkładzie łącznym nie można dokładnie wyznaczyć żadnego działania na tych zmiennych. Przyjmijmy, że interesuje nas rozkład sumy, to jest rozkład zmiennej $X + Y$. Bez żadnych dodatkowych założeń można podać jedynie ograniczenia na dystrybucję wyniku, podane przez G.D. Markowa (podrozdz. 3.5.4) w postaci dystrybucji: górnej $\overline{CDF}_{X+Y}$ oraz dolnej $\underline{CDF}_{X+Y}$. Znając pewną miarę zależności pomiędzy zmiennymi, można przybliżyć rozkład łączny wybraną kopułą z odpowiednio dobranym parametrem. W skrajnych przypadkach można zastanowić się, jak wyglądałby rozkład sumy, gdyby zmienne X i Y były skorelowane w maksymalnym stopniu dodatnio lub ujemnie, odpowiada to powiązaniu tych zmiennych kopułami M

i W . W tej sytuacji należy zastosować metody opisane w podrozdz. 3.5.3. Kopuły M i W można aproksymować kopułą Franka z parametrami dodatnimi (kopułę M) oraz ujemnymi (kopułę W). W pakiecie **PaCAL** możliwe jest wyznaczenie wszystkich powyższych rozkładów oraz ograniczeń. W przypadku absolutnie ciągłym wykorzystuje się zwykłą klasę `TwoVarsModel` w połączeniu z ustalonym modelem kopułowym. W pozostałych przypadkach należy użyć modeli związanych z kopułami W i M . Wyniki dla rozważanego przypadku przedstawiono na rys. 6.10.



Rys. 6.9. Rozkłady zmiennych: a) $Y \sim N(1.5, 0.45)$, b) $X \sim U[0, 1]$; pokazano funkcje gęstości oraz dystrybucyjne



Rys. 6.10. Skrajne możliwe wartości sumy zmiennych losowych $X + Y$: a) funkcje gęstości, b) dystrybucyjne; przedstawiono modele: maksymalnej korelacji dodatniej (M) i ujemnej (W), aproksymacje kopułą Franka z parametrami $\theta = 10$ (F_{+10}) i $\theta = -10$ (F_{-10}), ograniczenia Markowa (górne $\overline{CDF}$ (sup) i dolne $\underline{CDF}$ (inf)) oraz model zmiennych niezależnych (Π)

7. Podsumowanie

W pracy przedstawiono mechanizmy obliczeniowe na zmiennych losowych oparte na numerycznie dokładnych reprezentacjach funkcji gęstości wraz z praktyczną implementacją w postaci biblioteki **PaCAL**. Pakiet umożliwia wykonywanie operacji głównie na zmiennych niezależnych, jednak możliwe są również pewne typy obliczeń na zmiennych zależnych. Używając reprezentacji funkcji ciągłych wykorzystujących wielomiany Czebyszewa, uzyskano wysoką dokładność, bliską precyzji maszynowej, co było jednym z głównych założeń projektowych. Czas obliczeń, chociaż nie było to głównym priorytetem, jest również zadowalający, zwykle znacznie krótszy niż czas porównywalnych obliczeń symbolicznych. Obliczenie pojedynczej operacji jest zwykle rzędu sekund i zależy oczywiście od złożoności interpolatorów. Prezentowana biblioteka dostarcza ogólnych mechanizmów obliczeniowych, jednak w wielu przypadkach szczególnych może z powodzeniem konkurować ze standardowymi procedurami numerycznymi, przeznaczonymi do obliczania szczególnych typów rozkładów, co widać na przykładzie rozkładów niecentralnych. Główny wkład w doskonalenie metod i reprezentacji numerycznych w stosunku do działającego na podobnej zasadzie projektu **chebfun** polega na zastosowaniu odpowiednich transformacji zmiennych dla funkcji mających punkty osobliwe oraz funkcji określonych na przedziałach nieskończonych. Dzięki zastosowanym transformacjom udało się skutecznie rozszerzyć zakres stosowalności biblioteki na większość rozkładów ciągłych występujących w praktyce. Wiele dalszych przykładów można znaleźć na stronie <http://pacal.sourceforge.net/gallery/index.html>.

Główne kierunki dalszych badań i rozwoju projektu będą dotyczyć różnych aspektów obliczeń probabilistycznych, takich jak:

1. Rozwój metod wnioskowania w modelach graficznych wykorzystujących ciągle reprezentacje zmiennych wielowymiarowych. W tym zakresie punktem odniesienia są znane systemy wnioskowania probabilistycznego oparte na metodach Monte Carlo, jak **BUGS**. Wydaje się, że potencjał metod numerycznych w tym zakresie nie został jeszcze właściwie wykorzystany.

2. Poszerzenie obszaru zastosowań opisywanych metod obliczeniowych, w tym obliczeń na zmiennych zależnych oraz zmiennych wielowymiarowych (na przykład na zmienne wielowymiarowe określone na nośnikach nieskończonych).

3. Implementacje wybranych procedur wnioskowania statystycznego i wnioskowania bayesowskiego (jak analiza niezawodności czy analiza przeżycia).

Bibliografia

- [1] Abramowitz M., Stegun I., *Handbook of Mathematical Functions*, 5th ed., New York, Dover Publications, 1964, ISBN 978-0486612720.
- [2] Andrieu C., de Freitas N., Doucet A., Jordan M.I., An Introduction to MCMC for Machine Learning, *Machine Learning*, 2003, vol. 50, no. 1, s. 5–43.
- [3] Anthony M., Bartlett P.L., *Neural Network Learning: Theoretical Foundations*, Cambridge, Cambridge University Press, 1999, ISBN-13: 978-0521573535.
- [4] Ascher D., *Numerical Python*, tech. rep., UCRL-MA-128569, Lawrence Livermore National Laboratory, 2001.
- [5] Battles Z., Trefethen L.N., An extension of Matlab to continuous functions and operators, *SIAM J. Sci. Comp.*, 2004, vol. 25, no. 5, s. 1743–1770.
- [6] Behnel S., Bradshaw R., Citro C., Dalcin L., Seljebotn D., Smith K., Cython: The best of both worlds, *Computing in Science Engineering*, 2011, vol. 13, no. 2, s. 31–39.
- [7] Ben-Haim Y., *Information-gap decision theory: decisions under severe uncertainty*, San Diego, CA, Academic Press, 2006, ISBN 9780120882519.
- [8] Berleant D., Xie L., Zhang J., Statool: A tool for distribution envelope determination (denv), an interval-based algorithm for arithmetic on random variables, *Reliable Computing*, 2003, vol. 9, no. 2, s. 91–108.
- [9] Berrut J.P., Baltensperger R., Mittelmann H.D., Recent Developments in Barycentric Rational Interpolation, *International Series of Numerical Mathematics*, 2005, vol. 151, s. 27–51.
- [10] Berrut J.P., Trefethen L.N., Barycentric Lagrange Interpolation, *SIAM Rev.*, 2004, vol. 64, no. 3, s. 501–517.
- [11] Bishop C., *Pattern recognition and machine learning*, New York, Springer, 2006, ISBN 9780521642989.
- [12] Boyd J., *Chebyshev and Fourier Spectral Methods*, New York, Dover Publishers, 2001, ISBN 9780486411835.
- [13] Boyd J.E., Computing real roots of a polynomial in Chebyshev series form through subdivision, *Appl. Numer. Math.*, August 2006, vol. 56, s. 1077–1091.
- [14] Boyd J.P., Exponentially Convergent Fourier-Chebyshev Quadrature Schemes on Bounded and Infinite Intervals, *Journal of Scientific Computings*, 1987, vol. 2, no. 2, s. 99–109.
- [15] Butzer P., Jansche S., A direct approach to the mellin transform, *The Journal of Fourier Analysis and Applications*, 1997, vol. 3, no. 4, s. 325–376.

- [16] Cheeseman P., In defense of probability, in: *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, [b.w.], 1985, s. 1002–1009.
- [17] Cheney E.W., *Introduction to Approximation Theory*, Providence, Rhode Island, AMS Chelsea Publishing, 1982, ISBN 0-8218-1374-9.
- [18] Clauset A., Shalizi C., Newman M., Power-law distributions in empirical data, *SIAM Rev.*, 2009, vol. 51, s. 661–703.
- [19] Collins D., The relationship between Fourier and Mellin transforms, with applications to probability and stochastic processes, in: *LA-UR-11-03715*, 2011.
- [20] D'Agostini G., *Probability and Measurement Uncertainty in Physics - a Bayesian Primer*, [b.m.], Istituto Nazionale di Fisica Nucleare, 1995, ISBN 9780521592710.
- [21] Dagum P., Luby M., Approximating probabilistic inference in bayesian belief networks is np-hard, *Artif. Intell.*, 1993, vol. 60, s. 141–153.
- [22] Davis P.J., Rabinowitz P., *Methods of numerical integration*, London, Academic Press Inc., 1984, ISBN 9780122063503.
- [23] Day D., Romero L., Roots of polynomials expressed in terms of orthogonal polynomials, *SIAM J. Numer. Anal.*, May 2005, vol. 43, s. 1969–1987.
- [24] Druzdzel M.J., Intelligent decision support systems based on smile, *Software 2.0*, 2005, vol. 2, s. 12–33.
- [25] Dubois D., Prade H., *Possibility Theory*, New York, Plenum Press, 1998, ISBN 978-3-642-24739-2.
- [26] Edwards H.H., Taylor M.D., Characterizations of degree one bivariate measures of concordance, *J. Multivar. Anal.*, 2009, vol. 100, no. 8, s. 1777–1791.
- [27] Fejér L., On the infinite sequences arising in the theories of harmonic analysis, of interpolation, and of mechanical quadratures, *Bulletin of the American Mathematical Society*, 1933, vol. 39, no. 8, s. 521–534.
- [28] Feller W., *An Introduction to Probability Theory and Its Applications*, vol. 2, [b.m.], John Wiley & Sons Inc, 1971, ISBN 978-0471257080.
- [29] Ferson S., *RAMAS Risk Calc 4.0 Software: Risk Assessment with Uncertain Numbers*, Boca Raton, Florida, Lewis Publishers ed., 2002.
- [30] Ferson S., Kreinovich V., Ginzburg L., Myers D.S., Sentz K., Constructing Probability Boxes and Dempster-Shafer Structures, *SAND2002-4015*, 2003, no. 1, s. 1–143.
- [31] Fichtenhotz G.M., *Rachunek różniczkowy i całkowy*, vol. 2, Warszawa, PWN, 1963, ISBN 978-83-01-15063-1.
- [32] Fichtenhotz G.M., *Rachunek różniczkowy i całkowy*, vol. 3, Warszawa, PWN, 1963, ISBN 83-01-02175-6.
- [33] Fisz M., *Rachunek prawdopodobieństwa i statystyka matematyczna*, Warszawa, PWN, 1954, ISBN 9788301142926.
- [34] Floater M.S., Hormann K., Barycentric rational interpolation with no poles and high rates of approximation, *Numerische Mathematik*, 2007, vol. 107, no. 2, s. 315–331.
- [35] Folland G.B., *Real Analysis: Modern Techniques and Their Applications*, New York, John Wiley & Sons Inc, 1984, ISBN 978-0-471-31716-6.

- [36] Frank M.J., Nelsen R.B., Schweizer B., Best-possible bounds for the distribution of a sum – a problem of Kolmogorov, *Probability Theory and Related Fields*, 1987, vol. 74, s. 199–211.
- [37] Galassi M., Davies J., Theiler J., Gough B., Jungman G., Booth M., Rossi F., *Gnu Scientific Library: Reference Manual*, Network Theory Ltd., Feb. 2003, ISBN 978-0954612078.
- [38] Gander W., Gautschi W., Adaptive quadrature – revisited, *BIT*, 2000, vol. 40, no. 1, s. 084–101.
- [39] Glen A., Evans D., Leemis L., Appl: A probability programming language, *Am. Stat.*, May 2001, vol. 55, s. 156–166.
- [40] Hastings W., Monte Carlo sampling methods using Markov chains and their applications, *Biometrika*, 1970, s. 97–109.
- [41] Hofert M., Maechler M., Nested Archimedean Copulas Meet R: The nacopula Package, *Journal of Statistical Software*, 2011, vol. 39, no. 9, s. 1–20.
- [42] Hoffmann-Jørgensen J., *Probability with a View toward Statistics*, vol. 1, New York, Chapman and Hall, 1994, ISBN 978-0412052217.
- [43] Hoffmann-Jørgensen J., *Probability with a View toward Statistics*, vol. 2, New York, Chapman and Hall, 1994, ISBN 978-0412052217.
- [44] Hörmander L., *The Analysis of Linear Partial Differential Operators*, Springer-Verlag, 1990, ISBN 978-0-387-12139-0.
- [45] Hryniewicz O., *Nowoczesne metody statystycznego sterowania jakością*, Warszawa, Omnitex Press, 1996, 5-8114-0437-9.
- [46] Huard D., Évin G., Favre A.C., Bayesian copula selection, *Computational Statistics & Data Analysis*, 2006, vol. 51, no. 2, s. 809–822.
- [47] Hunter J.D., Matplotlib: A 2d graphics environment, *Computing In Science & Engineering*, 2007, vol. 9, no. 3, s. 90–95.
- [48] IEEE, *ANSI/IEEE 754-1985, Standard for Binary Floating-Point Arithmetic*, New York, 1985.
- [49] International Standards Organization (ISO), *Guide to the expression of uncertainty in measurement*, tech. rep., 1995.
- [50] Jakubowski J., Sztencel R., *Wstęp do teorii prawdopodobieństwa*, 2 ed., Warszawa, SCRIPT, 2001, ISBN 978-83-89716-01-9.
- [51] Jaroszewicz S., Korzeń M., Arithmetic Operations on Independent Random Variables: A Numerical Approach, *SIAM J. Sci. Comp.*, 2012, vol. 34, no. 4, s. A1241–A1265.
- [52] Jaroszewicz S., Korzeń M., *PaCAL – ProbAbilistic CALculator*, 2011, www.pacal.sf.net, dostęp 1.06.2012 r.
- [53] Jensen F.V., *Bayesian Networks and Decision Graphs*, Berlin, Springer, 2007, ISBN 9781441923943.
- [54] Jensen F.V., *Introduction to Bayesian Networks*, 1st ed., Secaucus, NJ, USA, Springer-Verlag, 1996, ISBN 0387915028.
- [55] Joe H., *Multivariate Models and Dependence Concepts*, Boca Roton, Chapman & Hall /CRC, 1997, ISBN 978-0412073311.

- [56] Johansson F., *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.14)*, 2010, <http://code.google.com/p/mpmath/>, dostęp 1.06.2012 r.
- [57] Johnson W.P., The curious history of faa' di bruno's formula, *The American Mathematical Monthly*, 2002, vol. 109, no. 3, s. 217–234.
- [58] Joint Committee for Guides in Metrology (JCGM), *Evaluation of measurement data – An introduction to the 'Guide to the expression of uncertainty in measurement' and related documents*, JCGM/WG 2 104:2008, Sèvres Cedex, France, 2009.
- [59] Joint Committee for Guides in Metrology (JCGM), *Evaluation of measurement data – Guide to the expression of uncertainty in measurement*, JCGM/WG 2 100:2008, Sèvres Cedex, France, 2008.
- [60] Joint Committee for Guides in Metrology (JCGM), *Evaluation of measurement data Supplement 1 to the "Guide to the expression of uncertainty in measurement" Propagation of distributions using a Monte Carlo method*, JCGM/WG 2 101:2008, Sèvres Cedex, France, 2008.
- [61] Jones E., Oliphant T., Peterson P., *SciPy: Open source scientific tools for Python*, tech. rep., <http://www.scipy.org/>, dostęp 1.06.2012 r, 2001.
- [62] Kaufmann A., Gupta M., *Introduction to Fuzzy Arithmetic*, [b.m.], Van Nostrand/Reinhold, 1985, ISBN 9780442230074.
- [63] Klir G.J., Folger T.A., *Fuzzy Sets, Uncertainty, and Information*, New Jersey, Prentice Hall International Inc., 1988, ISBN 9780133459845.
- [64] Knuth D.E., *Sztuka programowania*, vol. 2, Warszawa, WNT, 2002, ISBN 83-204-2553-0.
- [65] Knuth D.E., *Sztuka programowania*, vol. 3, Warszawa, WNT, 2002, ISBN 83-204-2554-9.
- [66] Kolassa J.E., Series approximation methods in statistics, *Lecture Notes in Statistics*, 2006, vol. 88, no. 1.
- [67] Koronacki J., Mielniczuk J., *Statystyka dla studentów kierunków technicznych i przyrodniczych*, Warszawa, WNT, 2004, ISBN 978-83-7011-859-4.
- [68] Korzeń M., Jaroszewicz S., *PaCal: a Python package for arithmetical computations with random variables*, praca w trakcie recenzji, 2011.
- [69] Kosiński W., On fuzzy number calculus, *Int. J. Appl. Math. Comput. Sci.*, 2006, vol. 16, no. 1, s. 51–57.
- [70] Krein M.G., Nudelman A.A., *The Markov moment problem and extremal problems*, [b.m.], American Mathematical Society, 1977, ISBN 9780821845004.
- [71] Krzyśko M., *Wykłady z teorii prawdopodobieństwa*, Warszawa, WNT, 2000, ISBN 978-83-204-2531-4.
- [72] Lebigot E.O., *Uncertainties: a Python package for calculations with uncertainties*, 2010, <http://packages.python.org/uncertainties>, dostęp 1.06.2012 r.
- [73] Leemis L.M., McQueston J.T., Univariate Distribution Relationships, *Am. Stat.*, 2008, vol. 62, no. 1, s. 45–53.
- [74] Leja F., *Funkcje zespolone*, Warszawa, PWN, 1996, ISBN 978-83-01-14948-2.
- [75] Li W., Hyman J.M., Computer arithmetic for probability distribution variables, *Reliability Engineering and System Safety*, 2004, vol. 85, s. 191–209.

- [76] Lunn D.J., Thomas A., Best N., Spiegelhalter D., WinBUGS – A Bayesian modelling framework: Concepts, structure, and extensibility, *Statistics and Computing*, 2000, vol. 10, s. 325–337.
- [77] Łęski J., *Systemy neuronowo-rozmyte*, Warszawa, WNT, 2008, ISBN 978-83-204-3329-6.
- [78] MacKay D.J.C., *Information theory, inference, and learning algorithms*, New York, Cambridge University Press, 2003, ISBN 9780521642989.
- [79] Mason J., Handscomb D., *Chebyshev Polynomials*, [b.m.], Chapman & Hall / CRC, 2003, ISBN 978-0849303555.
- [80] Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H., Teller E., Equation of state calculations by fast computing machines, *Journal of Chemical Physics*, 1953, vol. 21, s. 1087–1092.
- [81] Metropolis N., Ulam S., The monte carlo method, *J. Am. Stat. Assoc.*, 1949, vol. 44, s. 335.
- [82] Mikusiński P., Sherwood H., Taylor M.D., Probabilistic Interpretations of Copulas and Their Convex Sums, in: *Advances in Probability Distributions with Given Marginals*, eds. G. Dall’Aglio, S. Kotz, G. Salinetti, and M. Hazewinkel, vol. 67 of Mathematics and Its Applications, [b.m.], Springer Netherlands, 1991, s. 95–112.
- [83] Moore R.E., *Interval Analysis*, Englewood Cliffs N. J., Prentice-Hall, 1966, ISBN 9780898711615.
- [84] Moore R.E., *Interval Arithmetic and Automatic Error Analysis in Digital Computing*, PhD thesis, Department of Mathematics, Stanford University, Stanford, California, 1962.
- [85] Moore R.E., Kearfott R.B., Cloud M.J., *Introduction to Interval Analysis*, [b.m.], SIAM, 2009, ISBN 978-0-89871-669-6.
- [86] Näther W., Copulas and T-norms: Mathematical tools for combining probabilistic and fuzzy information, with application to error propagation and interaction, *Structural Safety*, 2010, vol. 32, no. 6, s. 366–371.
- [87] National Institute of Standards and Technology (NIST), *Digital Library of Mathematical Functions*, 2011, <http://dlmf.nist.gov/>, dostęp 1.06.2012 r.
- [88] Neal R.M., *Probabilistic Inference Using Markov Chain Monte Carlo Methods*, 1993, <http://cosmologist.info/Neal93>, dostęp 1.06.2012 r.
- [89] Nelsen R.B., *An Introduction to Copulas (Springer Series in Statistics)*, Secaucus, NJ, USA, Springer-Verlag New York, Inc., 2006, ISBN 9781441921093.
- [90] Neslehová J., On rank correlation measures for non-continuous random variables, *J. Multivar. Anal.*, 2007, vol. 98, no. 3, s. 544–567.
- [91] O’Hagan A., Buck C., Daneshkhah A., Eiser J., Garthwaite P., Jenkinson D., Oakley J., Rakow T., *Uncertain Judgements: Eliciting Experts’ Probabilities*, Statistics in Practice, West Sussex, England, John Wiley & Sons, 2006, ISBN 978-0-470-03330-2.
- [92] Osowski J., *Zarys rachunku operatorowego. Teoria i zastosowania w elektrotechnice*, Warszawa, WNT, 1972.
- [93] Owen D.B., The power of student’s t-test, *J. Am. Statist. Assoc.*, 1965, vol. 60, no. 309, s. 320–333.

- [94] Paszkowski S., *Zastosowania numeryczne wielomianów i szeregów Czebyszewa*, Warszawa, PWN, 1975, ISBN 978-0443068119.
- [95] Patil A., Huard D., Fonnesbeck C.J., Pymc: Bayesian stochastic modelling in python, *Journal of Statistical Software*, 2010, vol. 35, no. 4.
- [96] Pearl J., *Causality, models, reasoning and inference*, Cambridge, Cambridge University Press, 2000, ISBN 978-0521773621.
- [97] Pearl J., *Probabilistic Reasoning in Intelligent Systems*, San Mateo, CA, Morgan Kaufmann, 1988, ISBN 978-1558604797.
- [98] Piegat A., *Fuzzy modeling and control*, New York, Physica-Verlag, Springer, 2001, ISBN 978-3790813852.
- [99] Piessens R., deDoncker-Kapenga E., Uberhuber C., Kahaner D., *QUADPACK: A Subroutine Package for Automatic Integration*, Berlin, Springer-Verlag, 1983, ISBN 978-3540125532.
- [100] Platte R.B., Trefethen L.N., Kuijlaars A.B.J., Impossibility of fast stable approximation of analytic functions from equispaced samples, *SIAM Rev.*, 2011, vol. 53, no. 2, s. 308–318.
- [101] Pleśniak W., *Wykłady z teorii aproksymacji*, Wydawnictwo Uniwersytetu Jagiellońskiego, 2000, ISBN 83-233-1337-7.
- [102] Plummer M., *JAGS (Just Another Gibbs Sampler) 1.0.3 Universal*, 2010, <http://www-fis.iarc.fr/~martyn/software/jags/>, dostęp 1.06.2012 r.
- [103] Python Software Foundation, *Python Software, Version 2.7*, 2010, <http://www.python.org/>, dostęp 1.06.2012 r.
- [104] R Development Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011, ISBN 3-900051-07-0, <http://www.R-project.org/>, dostęp 1.06.2012 r.
- [105] Rump S., INTLAB – INTerval LABoratory, in: *Developments in Reliable Computing*, ed. T. Csendes, Kluwer Academic Publishers, Dordrecht, 1999, s. 77–104, <http://www.ti3.tu-harburg.de/rump/>, dostęp 1.06.2012 r..
- [106] SAS Institute, *SAS/QC(R) 9.3 User's Guide*, tech. rep., Cary, NC, USA, 2011, <http://support.sas.com/documentation/cdl/en/qcug/63964/PDF/default/qcug.pdf>, dostęp 1.06.2012 r.
- [107] Scarsini M., On measures of concordance, *Stochastica*, 1984, vol. 8, no. 3, s. 201–218.
- [108] Schmid F., Schmidt R., Blumentritt T., Gaïßer S., Ruppert M., Copula-based measures of multivariate association, in: *Copula Theory and Its Applications*, eds. P. Jaworski, F. Durante, W. K. Härdle, and T. Rychlik, vol. 198 of *Lecture Notes in Statistics*, Berlin, Heidelberg, Springer, 2010, s. 209–236.
- [109] Shafer G., *A Mathematical Theory of Evidence*, Princeton, Princeton University Press, 1976, ISBN 978-0691100425.
- [110] Siegrist K., *Virtual Laboratories in Probability and Statistics*, University of Alabama in Huntsville, <http://www.math.uah.edu/stat/>, dostęp 1.06.2012 r, 2001.
- [111] Sikorski R., *Rachunek różniczkowy i całkowy. Funkcje wielu zmiennych*, Warszawa, PWN, 1969, ISBN .

- [112] Spiegelhalter D., Thomas A., Best N., Lunn D., *WinBUGS User Manual*, 2003, <http://www.mrc-bsu.cam.ac.uk/bugs/>, dostęp 1.06.2012 r.
- [113] Springer M.D., *The Algebra of Random Variables*, New York, John Wiley & Sons, 1979, ISBN 978-0471014065.
- [114] Stoer J., *Wstęp do metod numerycznych*, vol. 1, Warszawa, PWN, 1979.
- [115] Sun Microsystems Inc., *C++ Interval Arithmetic Programming Reference*, tech. rep., 2002, <http://docs.oracle.com/cd/E19957-01/816-2465>, dostęp 1.06.2012 r.
- [116] Sun Microsystems Inc., *Numerical Computation Guide*, tech. rep., 2000, <http://docs.oracle.com/cd/E19957-01/806-3568/ncgTOC.html>, dostęp 1.06.2012 r.
- [117] SymPy Development Team, *SymPy: Python library for symbolic mathematics*, 2008, <http://www.sympy.org>, dostęp 1.06.2012 r.
- [118] Thompson J.R., Koronacki J., Nieckuła J., *Techniki zarządzania jakością: od Shewarta do metody „six Sigma”*, Warszawa, Akademicka Oficyna Wydawnicza EXIT, 2005, ISBN 978-83-7633-030-3.
- [119] Trefethen L.N., *Approximation theory and approximation practice*, University of Oxford, 2011.
- [120] Trefethen L.N., *Chebfun Version 4.0*. The Chebfun Development Team, 2011, <http://www.maths.ox.ac.uk/chebfun/>, dostęp 1.06.2012 r.
- [121] Trefethen L.N., Is Gauss Quadrature Better than Clenshaw–Curtis?, *SIAM Rev.*, 2008, vol. 50, s. 67–87.
- [122] Van Rossum G., *Python Library Reference*, Python Software Foundation, 2010, ISBN 978-0595136735.
- [123] Waldvogel J., Fast Construction of the Fejer and Clenshaw–Curtis Quadrature Rules, *BIT Numerical Mathematics*, 2003, vol. 43, no. 1, s. 001–018.
- [124] Williamson R.C., *Probabilistic Arithmetic*, PhD thesis, Department of Electrical Engineering, University of Queensland, St. Lucia, Queensland, Australia, 1989.
- [125] Wolfram Research, Inc., *Mathematica Edition: Version 8.0*, Champaign, Illinois, 2010.
- [126] Xiu D., Fast numerical methods for stochastic computations: A review, *Communications in Computational Physics*, 2009, vol. 5, no. 2-4, s. 242–272.
- [127] Zadeh L.A., Fuzzy sets, *Information and Control*, 1965, vol. 8, s. 338–353.
- [128] Zemke J., *b4m a free interval arithmetic toolbox for Matlab based on BAIS*, tech. rep., Technische Universität Hamburg, 1999, <http://w.ti3tu-harburg.de/~zemke/b4mindex.html>, dostęp 1.06.2012 r.
- [129] Zhang J., Berleant D., Arithmetic on Random Variables: Squeezing the Envelopes with New Joint Distribution Constraints, in: *Proceedings of the 4th International Symposium on Imprecise Probabilities and Their Applications*, eds. F., G. Cozman, R. Nau, and T. Seidenfeld, [b.m.], SIPTA, s. 416–422, 2005.
- [130] Zieliński R., Wiczorkowski R., *Komputerowe generatory liczb losowych*, Warszawa, WNT, 1997, ISBN 9788320421606.

Indeks

A

arytmetyka

interwałowa, 17

probabilistyczna, 16, 19, 25

rozmyta, 18

zmiennych losowych, 16, 19, 25

B

barycentric formula, 65

błąd

bezwzględny, 82

względny, 82

BMI, body mass index, 54

C

C-objętość, 46

CAS, 12, 124

ciężki ogon, 102

co-copula, 48

complementary cumulative distribution function, 48

Computer Algebra System, 12, 124

concordance function, 53

Cumulative Density Function, 24

D

dome, 47

dual copula, 47

dystrybuanta

dolna, 62

dopełniająca, 48

górną, 62

E

erf, error function, 13

F

$\mathcal{F}$, 7, 38–42, 89

FFT Fast Fourier Transform, 68

formuła barycentryczna, 65, 74, 84

Fréchet–Hoeffding bounds, 48

Frobenius companion matrix, 75

funkcja

całkowicie monotoniczna, 39

gęstości, 26

kawalkami gładka, 26–31

przeżycia, 48

z ciężkim ogonem, 102

H

Hamburger moment problem, 36

heavy-tailed distributions, 79

I

i.i.d. independent and identically distributed, 33

IEEE 754, 82

inference, 13

interpolacja

barycentryczna, 65, 66

trygonometryczna, 68

interpolator, 64

J

język

C/C++, 19, 106

Maple, 12, 126

Mathematica, 12, 101, 124, 126–128, 133, 134

Matlab, 19, 20, 29, 64, 74, 77, 79, 83, 89, 98, 101, 103, 126, 128, 133–135

Pyhton, 78
 Python, 20, 34, 70, 73, 78, 79, 90, 105,
 117, 124–126, 128, 154
 R, 125, 126, 133–136
 SAS, 125, 143
 JCGM, 18
 Joint Committee for Guides in Metrology, 18

K

klasa $\mathcal{F}$, 7, 38–42, 89
 ko-kopuła, 48
 konkordancja, 52
 kopuła, 46–48
 archimedesowa, 49
 Claytona, 50, 120
 dualna, 47
 Franka, 50, 120
 Gumbela, 50, 120
 M , 48
 W , 48
 Π , 48
 kwadratury Gaussa, 77, 78

L

linear opinion pool, 139
 linear score, 141
 logarithmic opinion pool, 139
 logarithmic score, 141

M

macierz stowarzyszona, 75
 marginalizacja, 63
 measure of concordance, 52
 metoda
 Clenshaw–Curtis, 64, 78
 Fejéra, 78
 miara zgodności, 52
 moc testu, 136
 moment zerowy, 123
 Monte Carlo, 14–15

N

norma

L^1 , 123
 trójkątna, 48

O

ograniczenia Frécheta–Hoeffdinga, 48

P

pakiet

APPL, 12, 126
b4m, 19
biblioteka arytmetyki interwałowej firmy Sun, 19
BUGS, 14, 125, 147
chebfun, 64, 71, 73, 87, 89, 90, 98–101,
 126, 128, 129, 147
Cython, 106
DSI Toolbox, 20
Fuzzy Logic Toolbox, 20
GSL, 77
INTLAB, 19
JAGS, 14
matplotlib, 106, 107, 114, 126
mpmath, 124
NumPy, 73, 74, 90, 98, 105, 126, 106
PaCAL, 9, 10, 15, 22, 23, 31, 35, 37,
 50, 57, 63, 64, 73, 78, 86–88, 90–92,
 98–101, 105, 106, 108, 112, 117–119,
 123–130, 132–135, 137, 141, 143–145,
 147, 163
pacal.integration, 91, 92
pacal.interpolation, 91
pacal.segments, 91, 92
pacal.stats.noncentral_distr, 132
polynomial.chebyshev, 73, 90
PyMC, 14, 125
QC, 143, 144
QUADPACK, 77, 103
Riskcalc, 20
SciPy, 106, 126
scipy.integrate, 79, 103
scipy.stats, 125, 133, 134
SMILE/Ginie, 14

- Statool**, 20
- SymPy**, 57, 106, 124, 126
- uncertainties**, 20, 118
- pierwiastki wielomianu, 75
- precyzja maszynowa, 82
- Probability Density Function, 24
- problem
 - Hamburgera, 36
 - Kołmogorowa, 61
- projekcja, 81
- propagacja niepewności, 16, 17
- propagation of uncertainty, 16
- punkt
 - osobliwy, 26, 85
 - potencjalnie osobliwy, 39, 85
- Q**
- quadratic score, 141
- R**
- ranked probability score, 141
- ratio distribution, 31, 130
- rozkład
 - Bernoulliego, 7, 112
 - beta, 7, 109, 118
 - Cauchy'ego, 7, 31, 101, 109
 - dwumianowy, 7, 112
 - F Snedecora, 7, 109
 - Frécheta, 7, 111
 - gamma, 7, 110
 - Gumbela, 7, 111
 - ilorazowy, 31, 130
 - jednostajny, 8, 35, 55, 101, 108, 127, 130, 131, 144, 145
 - Laplace'a, 7, 110
 - Levy'ego, 7, 101, 110
 - niecentralny beta, 7, 112, 133
 - niecentralny F, 7, 112, 133, 134
 - niecentralny T, 7, 111, 132–134
 - niecentralny χ^2 , 8, 111, 112, 133, 134
 - normalny, 7, 24, 45, 109, 111, 127, 144, 145
 - Pareto, 8, 102, 110
 - t-Studenta, 8, 109, 132
 - trapezowy, 8, 109
 - Wiebulla, 8, 111
 - wykładniczy, 7, 110
 - χ^2 , 7, 110–112
- rozstęp, 143
- Runge's phenomenon, 66
- S**
- segment, 85
- sieci bayesowskie, 13
- spherical score, 141
- statystyka
 - pozycyjna, 31, 33, 52
 - wnioskowanie, 135
- structural equations, 44
- Sun Microsystems, 19
- support, 17
- survival function, 48
- system operacyjny Windows, 106
- T**
- T-norma, 48
- transformata
 - Czebyszewa, 70
 - Fouriera, 35
- twierdzenie
 - o zamianie zmiennych, 54
 - Sklara, 46
 - Weierstrassa, 81
- W**
- wielomian
 - optymalny, 82
 - Czebyszewa, 68
 - całki, 74
 - pierwiastki, 76
 - pochodne, 75
 - Czebyszewa drugiego rodzaju, 68
 - Czebyszewa pierwszego rodzaju, 68
 - trygonometryczny, 67

wnioskowanie probabilistyczne, 21, 44, 45

współczynnik

Blomqvista, 54, 121

Giniego, 54

Kendalla, 54, 121

korelacji, 121

Spearmana, 54

Spermana, 121

zmienności, 137

wykładnik ogonkowy, 102

Z

zasada rozszerzenia Zadeha, 18

Numerical computations on random variables

Summary

The central theme of the monograph is the automation of numerical computations on random variables. Calculations in the model of independent variables and simpler models of dependent random variables were discussed thoroughly, including distributions defined with using of copulas or specific models, like the joint distribution of order statistics. The theoretical properties of computations on independent random variables were presented from the point of view of numerical implementation. The range of applicability of the methods presented is wide and it covers most of the distributions, occurring in practical problems, including: continuous variables, discrete variables and also mixed case.

The main idea presented in the paper consists in performing computations in a numerically accurate way, based on numerical representations of random variables and keeping the semantics similar to symbolic calculations. For this reason, a significant part of the paper concerns the practical implementation of numerical computations and representations of random variables. In this scope, barycentric interpolation, together with the method of approximation based on Chebyshev's polynomials was described. Furthermore, the methods of approximation and integration of functions defined on infinite domains and functions with singularities were presented. A lot of attention was given to the issue of the evaluation of the numerical accuracy. The practical implementation of the described methods was also presented – **PaCAL** library written in Python programming language. The library is publicly available at website: <http://pacal.sf.net>.

Performing computation with variables defined in an imprecise manner is of a great importance and appears in both practical and theoretical applications. Presented applications includes the computation of statistic distributions, propagation of uncertainties and measurement errors in metrology as well as the aggregation of experts judgements. The numerical accuracy of the presented approach is high, generally close to the machine precision. Despite of the fact that presented solutions are general, they are often more accurate than the generally applied specific routines, as for instance, in the case of the

implementation of non-central distributions. These distributions are computed frequently with a higher accuracy than the solutions implemented in the statistical libraries of software like Python, Matlab or R.

Numerische Berechnungen mit Zufallsvariablen Zusammenfassung

Das Leitthema der Monographie ist die Automatisierung von numerischen Berechnungen mit Zufallsvariablen. Es wurde ausführlich auf Berechnungen mit unabhängigen Variablen eingegangen und es wurden Berechnungen in einfachen Modellen von abhängigen Variablen dargestellt, die die mit Hilfe von Copulas erstellten Verteilungen umfassen, oder auch besondere Modelle, wie z.B. die Gesamtverteilung von Zufallsstatistiken. Es wurden theoretische Eigenschaften von Berechnungen mit unabhängigen Zufallsvariablen aus dem Gesichtspunkt der numerischen Implementierungen dargestellt. Der Einsatzbereich von präsentierten Methoden ist sehr breit und umfasst die Mehrheit von Verteilungen, die in praktischen Problemen auftreten, darin die Berechnungen von stetigen, diskreten und gemischten Zufallsvariablen.

Die in der Arbeit präsentierte Hauptidee besteht darin, die Berechnungen in einer numerisch genauen Weise durchzuführen und dabei auf numerischen Darstellungen von Zufallsvariablen zu basieren und unter Beibehaltung ähnlicher Semantik wie im Fall von symbolischen Berechnungen. Aus diesem Grund betrifft ein wesentlicher Teil der Arbeit praktische Implementierung von numerischen Mechanismen und Darstellungsarten von Zufallsvariablen. In diesem Bereich wurden die baryzentrische Interpolation und Approximationsmethoden beschrieben, die auf Tschebyscheff-Polynomen basieren. Darüber hinaus wurden die Methoden der Approximation und Integration von bestimmten Funktionen für unendliche Bereiche als auch von Funktionen mit Singularitäten dargestellt. Eine große Aufmerksamkeit schenkte man dem Problem der Bewertung der Genauigkeit von Berechnungen. Es wurde auch eine praktische Implementierung der beschriebenen Methoden vorgestellt – eine in der Python-Programmiersprache geschriebene **PaCAL**-Bibliothek. Diese Bibliothek ist öffentlich verfügbar unter Adresse <http://pocal.sf.net>.

Die Durchführung von Berechnungen mit Variablen, die in einer nicht präzisen Weise bestimmt wurden, hat eine große Bedeutung und erscheint in vielen sowohl praktischen als auch theoretischen Anwendungen. Der Anwendungsbereich umfasst die Bestimmung von Statistikverteilungen, Propagationen von Messunsicherheiten und Messfehlern

in der Metrologie oder auch die Aggregation von Expertenprognosen. Die Genauigkeit von erzielten Lösungen ist hoch und liegt gewöhnlich der maschinellen Präzision nahe. Obwohl die präsentierten Lösungen allgemeiner Natur sind, sind sie oft genauer als die allgemein eingesetzten Sondermethoden, wie zum Beispiel im Fall der Implementierung von Nichtzentralen-Verteilungen. Diese Verteilungen werden oft mit einer höheren Genauigkeit berechnet als Lösungen, die in statistischen Bibliotheken der Programme Python, Matlab oder R implementiert wurden.



Biblioteka Główna
Zachodniopomorskiego Uniwersytetu
Technologicznego w Szczecinie

W.141119



012-141119-00-0

ISBN 978-83-7663-126-4