universität wien

MAGISTERARBEIT

Titel der Magisterarbeit

„Kreuzvalidierung angewandt auf Approximate Bayesian Computation“

Verfasserin

Johanna Bertl, Bakk. rer. soc. oec.

angestrebter akademischer Grad

Magistra der Sozial- und Wirtschaftswissenschaften (Mag. rer. soc. oec.)

Wien, im August 2010

Studienkennzahl lt. Studienblatt: A 066 951

Dank

Besonders bedanken möchte ich mich bei Andreas Futschik, der mich auf die Idee gebracht hat, mich mit Populationsgenetik zu beschäftigen, und mich beim Schreiben dieser Diplomarbeit sehr unterstützt hat.

Ganz herzlich bedanken möchte ich mich auch bei meinem Freund Simon Eberle, der die Diplomarbeit Korrektur gelesen hat, und bei meinen Freundinnen Nikola Rüdisser und Ursula Bösch, die ihr Wissen über Genetik mit mir geteilt haben.

Abschließend möche ich mich noch bei meinen Eltern sehr herzlich bedanken, die mich beim Studium unterstützt haben - nicht zuletzt dadurch, dass sie mir immer alles zugetraut haben.

Inhaltsverzeichnis

Abbildungsverzeichnis

Einleitung

Mit immer schneller werdenen Computern können in der Statistik immer größere numerische Herausforderungen bewältigt werden. Ein Beispiel hierfür ist die Berechnung der a-posteriori-Verteilung eines Parameters in der Bayes-Statistik: In vielen praktischen Fällen kann sie nur mit iterativen numerischen Verfahren berechnet werden, deren Exaktheit mit der Anzahl der Iterationen zunimmt. Da diese Verfahren erst in den letzten zwei bis drei Jahrzehnten entwickelt und erforscht wurden, gibt es noch einige offene Fragen und viel Forschungsbedarf.

Eine numerische Methode zur Berechnung der a-posteriori-Verteilung ist Approximate Bayesian Computation (ABC). Sie kann verwendet werden, wenn die Likelihood nicht analytisch bestimmt werden kann. Die grundlegende Funktionsweise ist folgende: Es sei θ der Parameter des Modells $\mathcal{M}$, aus dem die Daten $\boldsymbol{X} = (X_1, \ldots, X_n)$ stammen.

Folgende drei Schritte werden m-mal wiederholt:

1. Simuliere θ^* aus der a-priori-Verteilung $\pi(\theta)$.

2. Simuliere Daten X^* aus $\mathcal{M}(\theta^*)$.

3. Berechne die Summary-Statistik S^* aus den simulierten Daten und vergleiche sie mit S, der Summary-Statistik aus den Ursprungsdaten. Wenn $d(S^*, S) \leq \epsilon$, akzeptiere θ^*, sonst, verwirf θ^*.

Aus den θ^*, die akzeptiert wurden, kann mit einer beliebigen Methode die a-posteriori-Dichte von θ geschätzt werden.

Die Beschreibung dieses Algorithmus wirft aber auch einige Fragen auf, die zwar in den letzten Jahren in vielen Arbeiten behandelt wurden, aber noch nicht endgültig geklärt sind:

- Welche Summary-Statistik(en) soll(en) verwendet werden und wieviele? Je mehr Summary-Statistiken, desto mehr Information ist über die Daten enthalten, allerdings wird bei höher dimensionalen Summary-Statistiken auch die Wahrscheinlichkeit, dass $d(S^*, S) \leq \epsilon$ zutrifft, ge-

ringer. Es gilt also, die Statistiken auszuwählen, die am meisten Information über die Daten enthalten.

- Wie soll ϵ gewählt werden? Je kleiner ϵ ist, desto näher ist die simulierte Verteilung der wahren a-posteriori-Verteilung, doch gleichzeitig wird die Anzahl der akzeptierten θ's immer kleiner, sodass die abschließende Dichteschätzung stark von den zufälligen Ergebnissen der Ziehungen von θ^* bzw. $\boldsymbol{X}^*$ abhängt.

- Welche Distanzfunktion $d()$ wird verwendet?

In dieser Diplomarbeit wird auf vor allem auf die Frage der Wahl von ϵ eingegangen.

Das bis dato wichtigste Anwendungsgebiet von ABC ist die Populationsgenetik. Der Coalescent-Prozess, der erstmals von Kingman (1982) definiert wurde, wird häufig verwendet, um die Entstehung der genetischen Variation innerhalb einer Population zu modellieren, da unterschiedliche populationsgenetische Modelle gegen den Coalescent-Prozess konvergieren. Seine Parameter können aber mit Maximum-Likelihood kaum oder nur mit enormem Aufwand geschätzt werden, sodass häufig Bayesianische Inferenz zur Anwendung kommt.

Kreuzvalidierung ist eine Methode zur Schätzung des Risikos eines Schätzers, die in der klassischen Statistik häufig zur Modellwahl oder zur Wahl eines statistischen Algorithmus verwendet wird. Sie kann aber auch in der Bayes-Statistik angewandt werden.

In dieser Arbeit wird anhand einer Simulation untersucht, ob Kreuzvalidierung zur Wahl von ϵ verwendet werden kann. Dazu werden DNA-Daten aus dem Standard-Coalescent-Prozess simuliert. Mittels ABC wird die a-posteriori-Verteilung des einzigen Parameters dieses Modells, der skalierten Mutationsrate, geschätzt.

In den Kapiteln 1, 2 und 3 werden ABC, Kreuzvalidierung und der Coalescent-Prozess jeweils getrennt von einander behandelt. Im Kapitel 4 wird die Simulationsstudie beschrieben, in der Kreuzvalidierung angewandt wird, um ϵ auszuwählen. Kapitel 5 präsentiert die Ergebnisse, die in der anschließenden Zusammenfassung interpretiert werden.

Teil I

Theorie

Kapitel 1

Approximate Bayesian Computation

1.1 Bayesianische Inferenz und Prognose

Es sei $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ ein Vektor von unabhängig identisch verteilten Zufallsvariablen, die Beobachtungen. X_i entstammt dem Modell $\mathcal{M}$ mit dem k-dimensionalen Parametervektor $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k) \in \boldsymbol{\Theta}$. $\boldsymbol{\pi}$ ist die k-dimensionale a-priori-Verteilung von $\boldsymbol{\theta}$.

Inferenz im bayesianischen Sinn basiert auf der a-posteriori-Verteilung $p(\boldsymbol{\theta}|\boldsymbol{X} = \boldsymbol{x})$ des Parameters $\boldsymbol{\theta}$. In diese fließt sowohl die a-priori-Verteilung des Parameters $\boldsymbol{\theta}$, die Vorinformationen über den Parameter enthält, die beispielsweise aus früheren Untersuchungen oder theoretischen Überlegungen gewonnen werden, als auch die Information, die die Daten geben, ein:

$$p(\boldsymbol{\theta}|\boldsymbol{X} = \boldsymbol{x}) = \frac{\pi(\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x})} \tag{1.1}$$

Je größer n, desto geringer ist der Einfluss der a-priori-Verteilung auf die a-posteriori-Verteilung.

Aus der a-posteriori-Verteilung von $\boldsymbol{\theta}$ kann auch eine Prognose-Verteilung für die Daten gewonnen werden. Wenn $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ die beobachteten Daten sind, kann ein Datenpunkt $\tilde{x}$ mit folgender Verteilung prognostiziert

werden (Gelman u. a., 2004, S. 8):

$$p(\tilde{x}|\boldsymbol{X}=\boldsymbol{x}) = \int_{\Theta} p(\tilde{x},\boldsymbol{\theta}|\boldsymbol{X}=\boldsymbol{x})d\boldsymbol{\theta} \tag{1.2}$$

$$= \int_{\Theta} p(\tilde{x}|\boldsymbol{\theta},\boldsymbol{X}=\boldsymbol{x})p(\boldsymbol{\theta}|\boldsymbol{X}=\boldsymbol{x})d\boldsymbol{\theta} \tag{1.3}$$

$$= \int_{\Theta} p(\tilde{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{X}=\boldsymbol{x})d\boldsymbol{\theta} \tag{1.4}$$

1.2 Numerische Methoden in der Bayes-Statistik

Bei der Berechnung der a-posteriori-Dichte (1.1) können analytische Schwierigkeiten auftreten:

Die Berechnung des Nenners $p(\boldsymbol{x})$, der das Integral über die a-posteriori-Dichte auf 1 normiert, kann die Berechnung mehrdimensionaler Integrale beinhalten:

$$p(\boldsymbol{x}) = \int_{\Theta} p(\boldsymbol{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta} \tag{1.5}$$

$$= \int_{\theta_1} \dots \int_{\theta_k} p(\boldsymbol{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\theta_k \dots d\theta_1 \tag{1.6}$$

Markov Chain Monte Carlo (MCMC) ist eine Simulationsmethode, mit Hilfe derer die Berechnung dieser Skalierungskonstante umgangen werden kann. Dabei wird eine Markov-Kette konstruiert, deren stationäre Verteilung die gesuchte a-posteriori-Verteilung ist.

In jedem Schritt der Simulation wird ein Wert der Markov-Kette für $\boldsymbol{\theta}^*$ simuliert. Die simulierten Werte nähern sich also immer mehr Beobachtungen aus der a-posteriori-Verteilung an. Es wird auch berücksichtigt, wie wahrscheinlich die Daten unter einem simulierten Parameter $\boldsymbol{\theta}^*$ sind, daher muss die Likelihood $p(\boldsymbol{x}|\boldsymbol{\theta}^*)$ in ausreichender Geschwindigkeit bestimmt werden können.

Die am häufigsten verwendeten Varianten des MCMC sind der Metropolis-Hastings-Algorithmus und der Gibbs-Sampler (siehe z. B. (Gelman u. a., 2004, Kapitel 11)).

Probleme kann auch die Berechnung der Likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ aufwerfen, die für die Berechnung des Zählers von (1.1) benötigt wird, wenn die Datenmenge zu groß oder das Modell sehr komplex ist (ein Beispiel dafür sind Coalescent-Modelle, wie in Kapitel 3 beschrieben). Ein Ansatz, mit dem die Berechnung

von Zähler und Nenner vermieden werden kann, ist es, eine Verteilung zu finden, die der a-posteriori-Verteilung ähnlich ist, und aus dieser Verteilung Werte für $\boldsymbol{\theta}$ zu simulieren. Diesem Ansatz folgt Approximate Bayesian Computation (ABC), wie es im Abschnitt 1.3 beschrieben wird.

1.2.1 Konjugierte Verteilungen

Durch die Verwendung von konjugierten a-priori-Verteilungen kann die a-posteriori-Verteilung ohne die aufwändige analytische oder numerische Berechnung (mehrdimensionaler) Integrale bestimmt werden (Carlin u. Louis, 1996, S. 32). Das ist jedoch nur in wenigen Fällen möglich.

Angenommen, die a-priori-Verteilung $\pi(\theta)$ des eindimensionalen Parameters θ gehört zu einer Verteilungsfamilie Π, die Likelihood $p(\boldsymbol{x}|\theta)$ zu einer Verteilungsfamilie $\mathcal{F}$. Wenn die Verteilungsfamilie Π zu $\mathcal{F}$ *konjugiert* ist, gehört die a-posteriori-Verteilung ebenfalls zu Π (Carlin u. Louis, 1996, Abschnitt 2.2.2).

Eine formale Definition ist beispielsweise in Gelman u. a. (2004, S. 41) zu finden:

Definition 1.1. *Es sei $\mathcal{F}$ eine Klasse von Verteilungen $p(\boldsymbol{x}|\theta)$ und Π eine Klasse von a-priori-Verteilungen von θ. Π ist genau dann eine konjugierte Klasse von Verteilungen für $\mathcal{F}$, wenn*

$$p(\theta|\boldsymbol{x}) \in \Pi \text{ für alle } p(\cdot|\theta) \in \mathcal{F} \text{ und } \pi(\cdot) \in \Pi \tag{1.7}$$

Diese Definition ist wenig brauchbar, wenn Π die Klasse aller Verteilungsfunktionen ist, denn dann ist Π immer konjugiert, ganz gleich, welche Likelihood $p(\boldsymbol{x}|\theta)$ verwendet wird. Sinnvoller ist es, eine Menge Π von parameterischen Verteilungsfunktionen mit derselben funktionalen Form zu betrachten. Die zugehörigen konjugierten Verteilungen werden als *natürliche* konjugierte Verteilungen bezeichnet. Die Berechnung der a-posteriori-Verteilung reduziert sich dann auf die Berechnung des Parameters θ der a-posteriori-Verteilung und ist (fast) immer möglich (Robert, 2007, S. 114).

Die Exponentialfamilie ist die einzige konjugierte Familie, daher wird Konjugiertheit manchmal über die Exponentialfamilie definiert (Gelman u. a., 2004, S. 42, siehe auch Definition 1.3).

In Fällen, in denen es keine einzelne konjugierte a-priori-Verteilung gibt, die die vorhandene Information über den Parameter θ angemessen widerspiegeln kann, kann eine Mischverteilung aus konjugierten a-priori-Verteilungen verwendet werden. Auch diese erleichtert die analytische Berechnung der a-posteriori-Verteilung (Carlin u. Louis, 1996, S. 33, Robert, 2007, S. 119). In

vielen Fällen ist es jedoch nicht möglich, die numerische Berechnung der a-posteriori-Verteilung zu umgehen, da es keine konjugierte a-priori-Verteilung gibt, die die a-priori-Informationen angemessen abbildet. Auch wenn eine nicht-informative a-priori-Verteilung gesucht wird, kann im Allgemeinen keine konjugierte Verteilung verwendet werden (Robert, 2007, S. 114).

Wenn es in Modellen mit mehrdimensionalem Parameter $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$ keine passende konjugierte a-priori-Verteilung gibt, kann für jeden einzelnen Parameter θ_i eine konjugierte a-priori-Verteilung spezifiziert werden. Die bedingten a-posteriori-Verteilungen entsprechen dann der konjugierten Form (Carlin u. Louis, 1996, S. 32).

1.3 Approximate Bayesian Computation (ABC)

1.3.1 Einleitung

Approximate Bayesian Computation (ABC) ist eine numerische Methode, mit der die a-posteriori-Verteilung $p(\boldsymbol{\theta}|\boldsymbol{X} = \boldsymbol{x})$ simuliert wird, wenn die Likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ nicht analytisch berechnet werden kann. Bis dato ist das Gebiet, in dem ABC am häufigsten zur Anwendung kommt, die statistische Populationsgenetik. Warum gerade bei populationsgenetischen Modellen die Likelihood selten analytisch bestimmt werden kann, wird in Kapitel 3 beschrieben.

1.3.2 Rejection-Algorithmen

Die grundlegende Funktionsweise eines Rejection-Algorithmus wird zum Beispiel bei Joyce u. Marjoram (2008) und Marjoram u. Tavaré (2006) beschrieben.

Unter der Annahme, dass die Daten $\boldsymbol{X}$ einem Modell $\mathcal{M}$ mit Parameter(n) $\boldsymbol{\theta}$ entstammen, die einer a-priori-Verteilung $\boldsymbol{\pi}$ folgen, besteht ein Rejection-Algorithmus aus folgenden Schritten (Die Realisierungen der Zufallsvariablen $\boldsymbol{X}$ werden mit $\boldsymbol{X}_{obs}$ bezeichnet):

Algorithmus 1.1.

1. *Simuliere* $\boldsymbol{\theta}^*$ *aus* $\boldsymbol{\pi}(\cdot)$.
2. *Simuliere Daten* $\boldsymbol{X}^*$ *aus dem Modell* $\mathcal{M}$ *mit dem simulierten Parametervektor* $\boldsymbol{\theta}^*$.
3. $\boldsymbol{\theta}^*$ *wird akzeptiert, wenn* $\boldsymbol{X}^* = \boldsymbol{X}_{obs}$, *sonst wird es verworfen.*

Die akzeptierten $\boldsymbol{\theta}^*$'s stellen „Beobachtungen" der a-posteriori-Verteilung $p(\boldsymbol{\theta}|\boldsymbol{X})$ dar.

Diese drei Schritte werden so oft wiederholt, bis eine so große Anzahl an $\boldsymbol{\theta}^*$'s akzeptiert wurde, dass die simulierte a-posteriori-Verteilung mit der gewünschten Genauigkeit geschätzt werden kann. Die einfachste Möglichkeit ist die empirische Verteilungsfunktion der $\boldsymbol{\theta}^*$'s, öfter werden aber Kerndichteschätzer verwendet.

Besonders bei hochdimensionalen Daten $\boldsymbol{X}$ ist die Wahrscheinlichkeit, dass die Bedingung $\boldsymbol{X}^* = \boldsymbol{X}_{obs}$ erfüllt werden kann, sehr klein. Daher sind sehr viele Wiederholungen des Rejection-Algorithmus notwendig, um eine ausreichend große Anzahl an $\boldsymbol{\theta}^*$'s zu erreichen.

Um die Wahrscheinlichkeit zu verringern, dass ein simulierter Parameter $\boldsymbol{\theta}^*$ verworfen wird, wird der Vergleich der simulierten Daten $\boldsymbol{X}^*$ mit den beobachteten Daten $\boldsymbol{X}_{obs}$ durch den Vergleich von Summary-Statistiken $\boldsymbol{S}(\boldsymbol{X})$ ersetzt. Diese sollen möglichst viel Information aus den Daten extrahieren, aber gleichzeitig nicht zu hochdimensional sein, da der Anteil der simulierten $\boldsymbol{\theta}^*$'s, die verworfen werden, sonst nicht ausreichend verringert werden kann (Genaueres zur Wahl der Summary-Statistiken findet sich in Kapitel 1.3.4).

Es sei $\boldsymbol{S} = \{S_1, S_2, \ldots, S_p\}$ eine Auswahl von Summary-Statistiken der Daten $\boldsymbol{X}$, $\boldsymbol{S}^* = \boldsymbol{S}(\boldsymbol{X}^*)$ und $\boldsymbol{S}_{obs} = \boldsymbol{S}(\boldsymbol{X}_{obs})$. Der dritte Schritt des oben beschriebenen Algorithmus wird durch folgenden ersetzt:

Algorithmus 1.2.

3' $\boldsymbol{\theta}^*$ wird akzeptiert, wenn $\boldsymbol{S}^* = \boldsymbol{S}_{obs}$, sonst wird es verworfen.

Die akzeptierten $\boldsymbol{\theta}^*$'s sind nun „Beobachtungen" der Verteilung $p(\boldsymbol{\theta}|\boldsymbol{S} = \boldsymbol{S}^*)$. Noch immer ist die Wahrscheinlichkeit, dass der Fall $\boldsymbol{S}^* = \boldsymbol{S}_{obs}$ eintritt, klein und der Rechenaufwand groß. Daher wird die genaue Übereinstimmung von $\boldsymbol{S}_{obs}$ und $\boldsymbol{S}^*$ dadurch ersetzt, dass $\boldsymbol{S}^*$ *in der Nähe* von $\boldsymbol{S}_{obs}$ liegen muss.

Es sei $\epsilon > 0$ eine beliebige kleine Konstante und $d(\boldsymbol{S}, \boldsymbol{S}^*)$ ein geeignetes Distanzmaß. Wieder wird der dritte Schritt des Algorithmus geändert:

Algorithmus 1.3.

3" $\boldsymbol{\theta}^*$ wird akzeptiert, wenn $d(\boldsymbol{S}_{obs}, \boldsymbol{S}^*) < \epsilon$, sonst wird es verworfen.

Die resultierenden $\boldsymbol{\theta}^*$ sind „Beobachtungen" der Verteilung $p(\boldsymbol{\theta}|d(\boldsymbol{S}, \boldsymbol{S}^*) < \epsilon)$. Wie gut diese Verteilung die a-posteriori-Verteilung $p(\boldsymbol{\theta}|\boldsymbol{X})$ annähert, kann nicht allgemein beantwortet werden.

Vor der Anwendung eines solchen Algorithmus müssen drei Fragen geklärt werden:

- Welche Summary-Statistiken werden verwendet?
- Welches Distanzmaß $d(\cdot)$ wird verwendet?
- Wie wird ϵ gewählt?

Auf die Frage der Summary-Statistiken wird in Abschnitt 1.3.4 eingegangen. Welche Distanzmaße für die Fragestellung geeignet sind, wird hier nicht behandelt. Die Wahl von ϵ hängt mit der Wahl der Summary-Statistiken zusammen. Es wird in den Abschnitten 1.3.3 und 1.3.4 darauf eingegangen, außerdem in der Simulation in Teil II.

1.3.3 Theoretische Fundierung

In Leuenberger u. a. (2009) wird gezeigt, warum $p(\boldsymbol{\theta}|d(\boldsymbol{S}, \boldsymbol{S}^*) < \epsilon)$ eine Annäherung an $p(\boldsymbol{\theta}|\boldsymbol{X})$ ist, wenn ϵ klein ist. Daraus wird auch ersichtlich, was Gründe für eine möglicherweise schlechte Annäherung sein können.

Angenommen, die Likelihood $p(\boldsymbol{s}|\boldsymbol{\theta})$ der Summary-Statistik $\boldsymbol{S}$ ist unter dem Modell $\mathcal{M}$ stetig in s und nicht null in der Umgebung der Beobachtungen $\boldsymbol{S}_{obs}$. Durch den Rejection-Algorithmus wird der Träger von $\boldsymbol{S}$ an den Grenzen, die durch $d(\boldsymbol{S}_{obs}, \boldsymbol{S})$ festgelegt werden, abgeschnitten. Aus der entstehenden Verteilung werden die Werte $\boldsymbol{S}^*$ simuliert. Daraus resultiert folgende Likelihood von $\boldsymbol{S}^*$:

$$p_\epsilon(\boldsymbol{s}|\boldsymbol{\theta}) = \mathbb{1}_{\boldsymbol{s}\in\mathcal{B}_\epsilon(\boldsymbol{s}_{obs})} \cdot p(\boldsymbol{s}|\boldsymbol{\theta}) \cdot \frac{1}{\int_{\mathcal{B}_\epsilon} p(\boldsymbol{s}|\boldsymbol{\theta}) d\boldsymbol{s}} \tag{1.8}$$

wobei $\mathcal{B}_\epsilon = \mathcal{B}_\epsilon(\boldsymbol{S}_{obs}) = \{\boldsymbol{s} \in \mathbb{R}^p | d(\boldsymbol{s}, \boldsymbol{S}_{obs}) < \epsilon\}$ die ϵ-Kugel in der Menge der Summary-Statistiken rund um die beobachtete Summary-Statistik $\boldsymbol{S}_{obs}$ ist (Leuenberger u. a., 2009, S. 2).

Die Indikatorfunktion $\mathbb{1}_{\boldsymbol{s}\in\mathcal{B}_\epsilon(\boldsymbol{s}_{obs})}$ schneidet die Likelihood am Rand der ϵ-Kugel ab. Der Term $p(\boldsymbol{s}|\boldsymbol{\theta})$ gibt die Form der Likelihood an. Sie ist identisch mit der Likelihood des nicht-abgeschnittenen Modelles, da durch $d(\boldsymbol{s}, \boldsymbol{s}^*) < \epsilon$ ja keine Veränderung der Form vorgenommen wird. Die Fläche unter der Dichte muss dennoch 1 sein, daher wird sie mit $1/\int_{\mathcal{B}_\epsilon} p(\boldsymbol{s}|\boldsymbol{\theta}) d\boldsymbol{s}$ normiert.

Die Dichte von $\boldsymbol{\theta}$ nach dem Rejection-Prozess wird wie folgt berechnet (um auf die Abhängigkeit von $\boldsymbol{S}$ von $\boldsymbol{\theta}$ hinzuweisen, wird in der folgenden Gleichung $\boldsymbol{S}(\boldsymbol{\theta})$ geschrieben):

$$\pi_\epsilon(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}|\boldsymbol{S}(\boldsymbol{\theta}) \in \mathcal{B}_\epsilon) = \frac{\pi(\boldsymbol{\theta}) P_{\boldsymbol{\theta}}(\boldsymbol{S}(\boldsymbol{\theta}) \in \mathcal{B}_\epsilon|\boldsymbol{\theta})}{P_{\boldsymbol{\theta}}(\boldsymbol{S}(\boldsymbol{\theta}) \in \mathcal{B}_\epsilon)} \tag{1.9}$$

$$= \frac{\pi(\boldsymbol{\theta}) \int_{\mathcal{B}_\epsilon} p(\boldsymbol{s}|\boldsymbol{\theta}) d\boldsymbol{s}}{\int_\Theta \pi(\boldsymbol{\theta}) \int_{\mathcal{B}_\epsilon} p(\boldsymbol{s}|\boldsymbol{\theta}) d\boldsymbol{s} d\boldsymbol{\theta}} \tag{1.10}$$

(Leuenberger u. a., 2009).

Daraus lässt sich nach Leuenberger u. a. (2009) zeigen, dass gilt:

$$\pi(\boldsymbol{\theta}|\boldsymbol{s}_{obs}) = \frac{p(\boldsymbol{s}_{obs}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int_\Theta p(\boldsymbol{s}_{obs}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}} = \frac{p_\epsilon(\boldsymbol{s}_{obs}|\boldsymbol{\theta})\pi_\epsilon(\boldsymbol{\theta})}{\int_\Theta p(\boldsymbol{s}_{obs}|\boldsymbol{\theta})\pi_\epsilon(\boldsymbol{\theta})d\boldsymbol{\theta}} \tag{1.11}$$

Mit dem Rejection-Algorithmus wird $\pi_\epsilon(\boldsymbol{\theta})$ geschätzt. Wenn ϵ klein ist, kann angenommen werden, dass $p(\boldsymbol{s}|\boldsymbol{\theta})$ auf $\mathcal{B}_\epsilon$ in etwa konstant ist. Daraus folgt:

$$p_\epsilon(\boldsymbol{s}|\boldsymbol{\theta}) \approx c \text{ für } \boldsymbol{s} \in \mathcal{B}_\epsilon(\boldsymbol{s}_{obs}) \tag{1.12}$$

wobei $c \in \mathbb{R}$ eine Konstante ist. Weiters folgt

$$\Rightarrow \int_\Theta p_\epsilon(\boldsymbol{s}|\boldsymbol{\theta})\pi_\epsilon(\boldsymbol{\theta})d\boldsymbol{\theta} \approx \int_\Theta c\pi_\epsilon(\boldsymbol{\theta})d\boldsymbol{\theta} = c\int_\Theta \pi_\epsilon(\boldsymbol{\theta})d\boldsymbol{\theta} = c \tag{1.13}$$

Aus 1.11 folgt dann:

$$\pi(\boldsymbol{\theta}|\boldsymbol{s}_{obs}) \approx \pi_\epsilon(\boldsymbol{\theta}) \tag{1.14}$$

Damit ist gerechtfertigt, dass $\pi_\epsilon(\boldsymbol{\theta})$ als Approximation für $\pi(\boldsymbol{\theta}|\boldsymbol{s}_{obs})$ verwendet wird.

Gleichzeitig werden die potenziellen Schwächen dieser Approximation sichtbar:

- Die Approximation hat nur Gültigkeit, wenn ϵ klein ist. Je höherdimensional die Summary-Statistik, desto größer muss ϵ aber gemacht werden, um die Rate der akzeptierten $\boldsymbol{\theta}^*$ nicht zu klein werden zu lassen. Bei komplexeren Modellen sind Summary-Statistiken der Dimension $p = 50$ nicht ungewöhnlich (Leuenberger u. a., 2009, S. 3).
- Die Variation von $p(\boldsymbol{s}|\boldsymbol{\theta})$ auf $\mathcal{B}_\epsilon$ wird nicht berücksichtigt.
- Ein kleines ϵ verringert die Anzahl an akzeptierten $\boldsymbol{\theta}^*$. Die Schätzung der a-posteriori-Verteilung wird daher stark vom zufälligen Ergebnis der Simulation beeinflusst (Joyce u. Marjoram, 2008, S. 2).

1.3.4 Wahl der Summary-Statistiken

$\boldsymbol{X} \in \mathbb{R}^n$ sind Beobachtungen, die dem Modell $\mathcal{M}$ mit Parameter $\boldsymbol{\theta}$ entstammen.

Definition 1.2. *Es sei $\boldsymbol{S} = \boldsymbol{S}(\boldsymbol{X})$ eine Statistik der Daten $\boldsymbol{X}$. $\boldsymbol{S}(\boldsymbol{X})$ ist genau dann suffizient für den Parameter $\boldsymbol{\theta}$, wenn $p(\boldsymbol{X}|\boldsymbol{S}(\boldsymbol{X}) = \boldsymbol{s})$ unabhängig von $\boldsymbol{\theta}$ ist (Bickel u. Doksum, 2006, S. 42).*

Das heißt, wenn die Realisierung $\boldsymbol{s}$ einer suffizienten Statistik $\boldsymbol{S}(\boldsymbol{X})$ bekannt ist, können die Daten $\boldsymbol{X}$ keine zusätzlichen Informationen über $\boldsymbol{\theta}$ liefern; die ganze Information, die in den Daten über $\boldsymbol{\theta}$ enthalten ist, wird in $\boldsymbol{S}(\boldsymbol{X})$ abgebildet.

Für den Algorithmus 1.2 heißt das, dass, wenn die verwendete Statistik $\boldsymbol{S}$ suffizient für $\boldsymbol{\theta}$ ist, $p(\boldsymbol{\theta}|\boldsymbol{S}(\boldsymbol{X})) = p(\boldsymbol{\theta}|\boldsymbol{X})$ gilt. Es geht also keine Information verloren, wenn anstatt der gesamten Daten die suffizienten Statistiken verwendet werden.

Definition 1.3. *Eine Familie von Verteilungen* $\{\mathcal{F}_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}, \Theta \subseteq \mathbb{R}^k$ *ist eine* k*-parametrige Exponentialfamilie, wenn reell-wertige Funktionen* $\eta_1, \ldots, \eta_k$ *und* B *von* $\boldsymbol{\theta}$ *und reell-wertige Funktionen* $T_1, \ldots, T_k, h$ *auf* $\mathbb{R}^n$ *existieren, sodass für die Dichte von* $\mathcal{F}_{\boldsymbol{\theta}}$ *gilt:*

$$p(x, \boldsymbol{\theta}) = h(x) \exp\left(\sum_{j=1}^{k} \eta_j(\boldsymbol{\theta}) T_j(x) - B(\boldsymbol{\theta})\right), x \in \mathbb{R}^n \tag{1.15}$$

(Bickel u. Doksum, 2006, S. 53).

Der Vektor $\boldsymbol{T}(\boldsymbol{X}) = (T_1(\boldsymbol{X}), \ldots, T_k(\boldsymbol{X}))$ ist suffizient für $\boldsymbol{\theta}$ und heißt *natürliche suffiziente Statistik* von $\mathcal{F}_{\boldsymbol{\theta}}$ (Bickel u. Doksum, 2006, S. 54).

Satz 1.1 (Pitman, Koopman, Darmois)**.** *Wenn* $p(x, \boldsymbol{\theta})$ *eine Familie von Dichten mit dem* k*-dimensionalen Parameter* $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$ *ist, dann hat die Familie der* n*-dimensionalen Verteilungen* $\prod_{i=1}^{n} p(x_i, \boldsymbol{\theta})$ *genau dann eine suffiziente Statistik der Dimension* $s < n$*, wenn* $p(x, \boldsymbol{\theta})$ *zur Exponentialfamilie gehört (Barankin u. Maitra, 1963, S. 217).*

Das heißt, Verteilungen, die nicht zur Exponentialfamilie gehören, haben keine suffizienten Statistiken mit kleineren Dimensionen als n. Eine suffiziente Statistik sind dann zum Beispiel die Beobachtungen selbst.

In solchen Fällen ist die Auswahl und die richtige Anzahl der Summary-Statistiken ein nicht-triviales Problem, mit dem sich Joyce u. Marjoram (2008) und Blum (2010) beschäftigt haben.

Natürlich kann man umso mehr Informationen aus den Daten über $\boldsymbol{\theta}$ gewinnen, je mehr Statistiken $S_1, S_2, \ldots$ man verwendet; je größer die Anzahl der Statistiken, desto geringer jedoch die Wahrscheinlichkeit, dass $d(\boldsymbol{S}, \boldsymbol{S}^*) < \epsilon$, und desto größer daher der Rechenaufwand. ϵ kann auch nicht beliebig vergrößert werden, da die Annäherung von $p(\boldsymbol{\theta}|d(\boldsymbol{S}, \boldsymbol{S}^*) < \epsilon)$ an $p(\boldsymbol{\theta}|\boldsymbol{X})$ mit größerem ϵ schlechter wird (siehe Abschnitt 1.3.3). Es muss also eine Methode gefunden werden, die Statistiken, die viel Information über die Daten enthalten, auszuwählen.

Joyce u. Marjoram (2008) schlagen „Approximate Sufficiency“ vor: Es wird eine Maßzahl definiert, die den Informationsgehalt einer neuen Statistik relativ zu einer schon vorhandenen Liste von Statistiken angibt. Eine neue Statistik wird in die Liste aufgenommen, solange die Maßzahl über einem festgesetzten Mindestwert liegt.

In Blum (2010) wird eine Methode vorgeschlagen, mit der nicht nur eine Auswahl aus einer Liste von Summary-Statistiken getroffen werden kann, sondern auch bestimmt werden kann, ob eine Summary-Statistik logarithmiert werden soll. Weiters kann die Methode verwendet werden, um den Anteil p_ϵ der simulierten $\boldsymbol{\theta}^*$, die akzeptiert werden, festzulegen. Dazu verwendet Blum die Regressionskorrektur von Beaumont u. a. (2002), die in Abschnitt 1.3.5 beschrieben wird, allerdings mit einem Bayesianischen Regressionsmodell. Anhand von Beispielen wird gezeigt, dass die „evidence function“, die aus diesem Modell berechnet wird, dazu verwendet werden kann, den ABC-Algorithmus zu parametrisieren.

1.3.5 Verbesserungen mittels Regression

Es sei θ_i^* der Parameter aus der i-ten Simulation, $i = 1, \ldots, m$, und $\boldsymbol{s}_i^*$ der aus den mit θ_i^* simulierten Daten berechnete Vektor von Summary-Statistiken.

Um ϵ größer wählen zu können und damit die mögliche Anzahl der Summary-Statistiken zu erhöhen, entwickelt Beaumont u. a. (2002) eine Variante von ABC, die den Abstand einer simulierten Summary-Statistik $\boldsymbol{S}_i^*$ zu dem beobachteten Wert $\boldsymbol{S}_{obs}$ berücksichtigt.

Zur Erklärung wird vorerst ein lineares Regressionsmodell verwendet, das anschließend zu einem lokal-linearen Modell erweitert wird. In diesem Abschnitt wird von einem eindimensionalen Parameter θ ausgegangen, das Modell kann jedoch für einen multivariaten Parameter adaptiert werden, indem das univariate Regressionsmodell durch ein multivariates ersetzt wird.

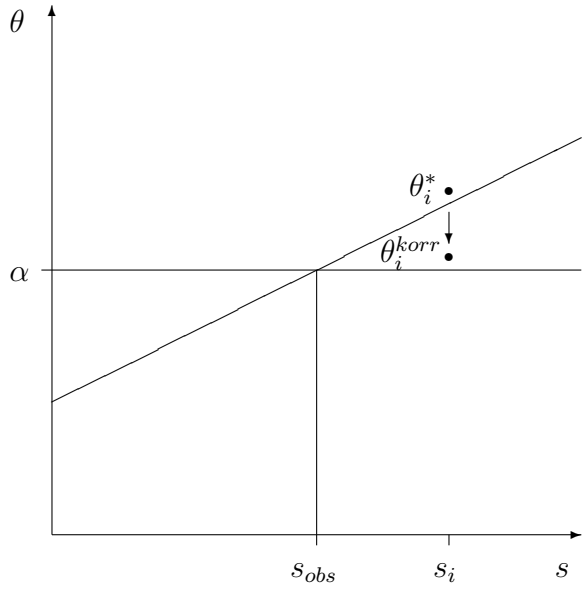
Die Regressionsgleichung lautet:

$$\theta_i^* = \alpha + (\boldsymbol{s}_i^* - \boldsymbol{s}_{obs})'\beta + \zeta_i, i = 1, \ldots, m \tag{1.16}$$

Es wird die Annahme getroffen, dass die ζ_i unkorreliert sind mit Mittelwert 0 und homogenen Varianzen. α und β werden mittels OLS geschätzt.

Wenn $\boldsymbol{s}_i = \boldsymbol{s}_{obs}$, dann stammen die θ_i aus der a-posteriori-Verteilung $\pi(\theta|\boldsymbol{S} = \boldsymbol{s}_{obs})$ und es gilt $\alpha = E(\theta|\boldsymbol{S} = \boldsymbol{s}_{obs})$. Das trifft aber nur auf ganz wenige oder gar keine der Simulationen zu. Wenn sich der Parameter β von null unterscheidet, weichen die Werte der θ_i^* mit $\boldsymbol{s}_i^* \neq \boldsymbol{s}_{obs}$ von den Beobachtungen θ_i aus der a-posteriori-Verteilung $\pi(\theta|\boldsymbol{S} = \boldsymbol{s}_{obs})$ ab. Diese Abweichung wird von Beaumont u. a. (2002, S. 2027) folgendermaßen korrigiert:

$$\theta_i^{korr} = \theta_i^* - (\boldsymbol{s}_i^* - \boldsymbol{s}_{obs})'\hat{\beta}_{OLS} \tag{1.17}$$



Die θ_i^{korr} bilden nun Beobachtungen aus der a-posteriori-Verteilung $\pi(\theta|\boldsymbol{S} = \boldsymbol{s}_{obs})$.

Zusätzlich zu dieser Korrektur wird der Einfluss der weit von $\boldsymbol{s}_{obs}$ entfernt liegenden $\boldsymbol{s}_i^*$ und der dazugehörenden θ_i^* auf die Regressionsgleichung mittels Gewichtung verringert. Beaumont u. a. (2002) verwenden ein lokal-lineares Regressionsmodell, in dem nicht, wie bei der OLS-Schätzung

$$\sum_{i=1}^{m} (\theta_i^* - \alpha - (\boldsymbol{s}_i^* - \boldsymbol{s}_{obs})'\beta)^2, \tag{1.18}$$

sondern

$$\sum_{i=1}^{m} (\theta_i^* - \alpha - (\boldsymbol{s}_i^* - \boldsymbol{s}_{obs})'\beta)^2 K_\epsilon(d(\boldsymbol{s}_i^*, \boldsymbol{s}_{obs})) \tag{1.19}$$

minimiert wird, wobei $K_\epsilon(\cdot)$ der Epanechnikov-Kern ist.

Der Epanechnikov-Kern hat folgende Form (siehe auch Abbildung 1.1):

$$K_\epsilon(t) = \begin{cases} c\epsilon^{-1}(1 - (t/\epsilon)^2), & t \leq \epsilon \\ 0, & t > \epsilon \end{cases} \tag{1.20}$$

c ist eine Konstante, die den Kern auf die Fläche eins normiert.

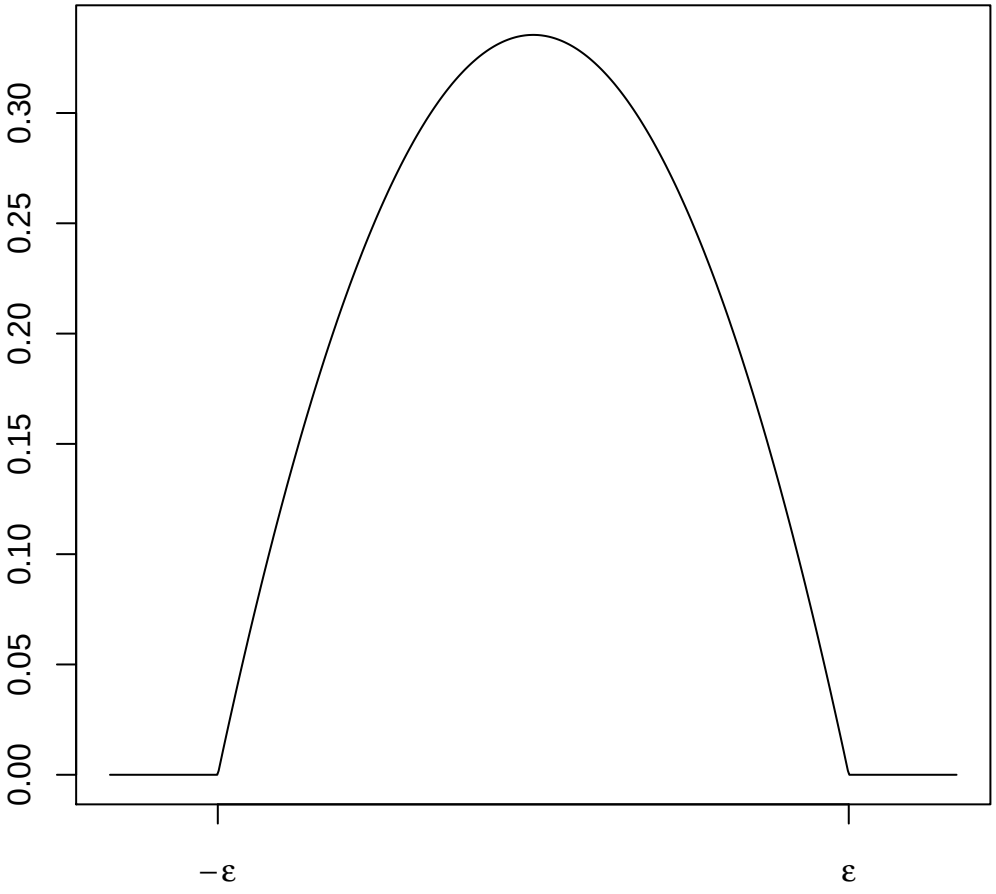


Abbildung 1.1: Epanechnikov-Kern

Es kann auch eine andere Kernfunktion verwendet werden, der Epanechnikov-Kern ist aber gut geeignet die Funktionsweise der Gewichtung im Zusammenhang mit dem Rejection-Algorithmus zu erklären.

Dass der Epanechnikov-Kern nur auf $[-\epsilon, \epsilon]$ größer null ist, zeigt den Zusammenhang mit dem Rejection-Algorithmus: Beobachtungen, die außerhalb liegen, werden nicht zur Schätzung der Regressionsparameter verwendet. Das sind die Simulationsergebnisse, die vom Rejection-Algorithmus verworfen werden. Die übrigen werden je nach Funktionswert des Kerns gewichtet (siehe Abbildung 1.1). Ohne Gewichtung entspricht der Rejection-Algorithmus mit der oben beschriebenen Korrektur der lokal-linearen Regression mit Indikator-Kern

$$I_\epsilon(t) = \begin{cases} 1, & t \leq \epsilon \\ 0, & t > \epsilon \end{cases} \tag{1.21}$$

Die Bezeichnung „lokal-lineare Regression“ kommt daher, dass angenommen wird, dass der lineare Zusammenhang nur auf der $\epsilon-$Kugel gilt. In den meisten Anwendungen ist sie so klein, dass sie nur einen Teil der Daten enthält. Die ϵ-Kugel wird dann wie bei der Kerndichteschätzung wie ein „Fenster“ an den Daten entlang „verschoben“ und die Parameterschätzer werden an jedem Datenpunkt einzeln geschätzt. Daraus ergibt sich eine nichtparametrische Kurvenschätzung. Hier ist die ϵ-Umgebung um $\boldsymbol{s}_{obs}$ zentriert und wird nicht verschoben. Es handelt sich also um eine gewichtete lineare Regression.

Eine Variante dieses Modells wird in Blum u. François (2010) vorgeschlagen: Die Korrektur soll flexibler sein und mittels eines nicht-linearen Regressionsmodells erfolgen.

1.3.6 Beispiel: Normalverteilung

Die Normalverteilung gehört zur Exponentialfamilie, daher gilt: Wenn die Beobachtungen $\boldsymbol{X} = (X_1, \ldots, X_n)$ normalverteilt sind, kann eine konjugierte Verteilung als a-priori-Verteilung gewählt werden, sodass die a-posteriori-Verteilung dieselbe funktionale Form, jedoch andere Parameter hat (siehe Kapitel 1.2.1). Diese hängen von den Daten und den Parametern der a-priori-Verteilung und der Verteilung der Daten ab. Im Fall von normalverteilten Daten handelt es sich auch bei der konjugierten Verteilung um eine Normalverteilung (Bickel u. Doksum, 2006, S. 63–64).

Es seien $\boldsymbol{X} = (X_1, \ldots, X_n)$ ein-dimensionale Beobachtungen mit $X_i \sim \mathcal{N}(\mu, \sigma^2)$, wobei die Varianz σ^2 bekannt ist und $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$. Weiters sei $\bar{X} = \sum_{i=1}^{n} X_i$ das arithmetische Mittel von $\boldsymbol{X}$, eine suffiziente Statistik für μ, wenn σ bekannt ist.

Die a-posteriori-Verteilung ist die Normalverteilung $\mathcal{N}(\mu_1, \sigma_1^2)$, wobei

$$\mu_1 = \left(\frac{\sigma^2}{\sigma_0^2} + n\right)^{-1} \left(\sum_{i=1}^{n} x_i + \frac{\mu_0 \sigma^2}{\sigma_0^2}\right) \tag{1.22}$$

$$\sigma_1^2 = \left(\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}\right)^{-1} \tag{1.23}$$

(Bickel u. Doksum, 2006, S. 63–64).

Wird ein ABC-Rejection-Algorithmus vom Typ 1.3 auf dieses Beispiel angewandt, sieht der Algorithmus für $i = 1, \ldots, m$ Wiederholungen folgendermaßen aus:

Algorithmus 1.4.

1. *Simuliere* μ_i^* *aus* $\mathcal{N}(\mu_0, \sigma_0^2)$.
2. *Simuliere Daten* $\boldsymbol{X}_i^* = (X_{i,1}^*, \ldots, X_{i,n}^*)$ *aus* $\mathcal{N}(\mu_i^*, \sigma^2)$.
3. *Berechne* $\bar{X}_i^* = \frac{1}{n}\sum_{j=1}^{n} X_{i,j}^*$. μ_i^* *wird akzeptiert, wenn* $\left|\bar{X}_i^* - \bar{X}\right| < \epsilon$, *sonst wird es verworfen.*

Wie im Kapitel 1.3.3 beschrieben, hat μ_i^* die Dichte $\pi_\epsilon(\mu)$. Diese ist nach Gleichung 1.9:

$$\pi_\epsilon(\mu) = \frac{\pi(\mu) \int_{\bar{x}_{obs}-\epsilon}^{\bar{x}_{obs}+\epsilon} p(\bar{x}|\mu) d\bar{x}}{\int_{-\infty}^{\infty} \pi(\mu) \int_{\bar{x}_{obs}-\epsilon}^{\bar{x}_{obs}+\epsilon} p(\bar{x}|\mu) d\bar{x} d\mu} \tag{1.24}$$

Es sei $\phi_{\mu,\sigma^2}(x)$ die Dichte der Normalverteilung mit Parametern μ und σ^2. Dann gilt:

$$\pi_\epsilon(\mu) = \frac{\phi_{\mu_0,\sigma_0^2}(\mu) \int_{\bar{x}_{obs}-\epsilon}^{\bar{x}_{obs}+\epsilon} \phi_{\mu,\sigma^2/n}(\bar{x}|\mu) d\bar{x}}{\int_{-\infty}^{\infty} \phi_{\mu_0,\sigma_0^2}(\mu) \int_{\bar{x}_{obs}-\epsilon}^{\bar{x}_{obs}+\epsilon} \phi_{\mu,\sigma^2/n}(\bar{x}|\mu) d\bar{x} d\mu} \tag{1.25}$$

$\pi_\epsilon(\mu)$ hängt neben den bekannten Parametern μ_0, σ_0^2, σ_2 und n von $\bar{x}_{obs}$ und der Wahl von ϵ ab. Es ist keine Normalverteilungsdichte:

$$\pi_\epsilon(\mu) = \frac{\phi_{\mu_0,\sigma_0^2}(\mu) f_{\sigma^2,n}(\mu; \bar{x}_{obs}, \epsilon)}{c_{\mu_0,\sigma_0^2,\sigma^2,n}(\bar{x}_{obs}, \epsilon)} \tag{1.26}$$

mit

$$f_{\sigma^2,n}(\mu; \bar{x}_{obs}, \epsilon) = \int_{\bar{x}_{obs}-\epsilon}^{\bar{x}_{obs}+\epsilon} \phi_{\mu,\sigma^2/n}(\bar{x}|\mu) d\bar{x} \tag{1.27}$$

$$c_{\mu_0,\sigma_0^2,\sigma^2,n}(\bar{x}_{obs}, \epsilon) = \int_{-\infty}^{\infty} \phi_{\mu_0,\sigma_0^2}(\mu) \int_{\bar{x}_{obs}-\epsilon}^{\bar{x}_{obs}+\epsilon} \phi_{\mu,\sigma^2/n}(\bar{x}|\mu) d\bar{x} d\mu \tag{1.28}$$

Sowohl $f_{\sigma^2,n}(\mu; \bar{x}_{obs}, \epsilon)$ als auch $c_{\mu_0,\sigma_0^2,\sigma^2,n}(\bar{x}_{obs}, \epsilon)$ können nicht analytisch ausgewertet werden, da sie Integrale über die Dichte der Normalverteilung enthalten. Die Abweichung von $\pi_\epsilon(\mu)$ von der wahren a-posteriori-Verteilung kann daher nur für einen fixen Datensatz $\boldsymbol{x}$ und verschiedene Werte von ϵ und nicht allgemein betrachtet werden.

Wie sich die wahre a-posteriori-Verteilung von der theoretischen ABC-a-posteriori-Verteilung unterscheidet, zeigen folgende Beispiele:

Beispiel 1: Als a-priori-Verteilung wird die Standardnormalverteilung angenommen. Der Parameter μ der Verteilung der Daten wird nicht zufällig gezogen, sondern bei 0 festgelegt. Als Daten werden 20 zufällige Werte aus der Standardnormalverteilung gezogen. Die wahre a-posteriori-Verteilung wird nach den Gleichungen 1.22 bzw. 1.23 und die ABC-a-posteriori-Verteilung nach 1.26 berechnet. Die numerische Integration von Gleichung 1.28 wird in R 2.11.1 mit der Funktion `cuhre` aus dem Package `R2Cuba` durchgeführt.

Deskriptive Statistiken der generierten Daten:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Var.
-2.21	-0.38	0.36	0.19	0.76	1.59	0.83

In Abbildung 1.2 ist zu sehen, wie sehr sich die ABC-a-posteriori-Dichte von der wahren a-posteriori-Dichte unterscheidet, wenn ϵ größer gewählt wird. Da die ABC-a-posteriori-Dichte theoretisch berechnet wurde und nicht durch Simulation, ist nicht zu sehen, dass ein kleineres ϵ gleichzeitig die Rate an akzeptierten μ^*'s verringert und damit die Dichteschätzung stark von den zufälligen Werten von μ^* abhängt. Das heißt, die Annäherung der ABC-a-posteriori-Dichte an die wahre a-posteriori-Dichte ist in einem Beispiel mit Simulation nicht so gut wie in diesem theoretischen Beispiel.

Beispiel 2: Als a-priori-Verteilung wird wieder die Standardnormalverteilung angenommen. Der Parameter μ wird aber bei 1 festgelegt. Die Anzahl der Daten ist wieder 20 und $\sigma = 1$. Wie in Beispiel 1 werden die wahre a-posteriori-Dichte und die theoretische ABC-a-posteriori-Dichte berechnet.

Deskriptive Statistiken der generierten Daten:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Var.
-1.28	0.39	1.30	1.14	1.77	3.20	1.30

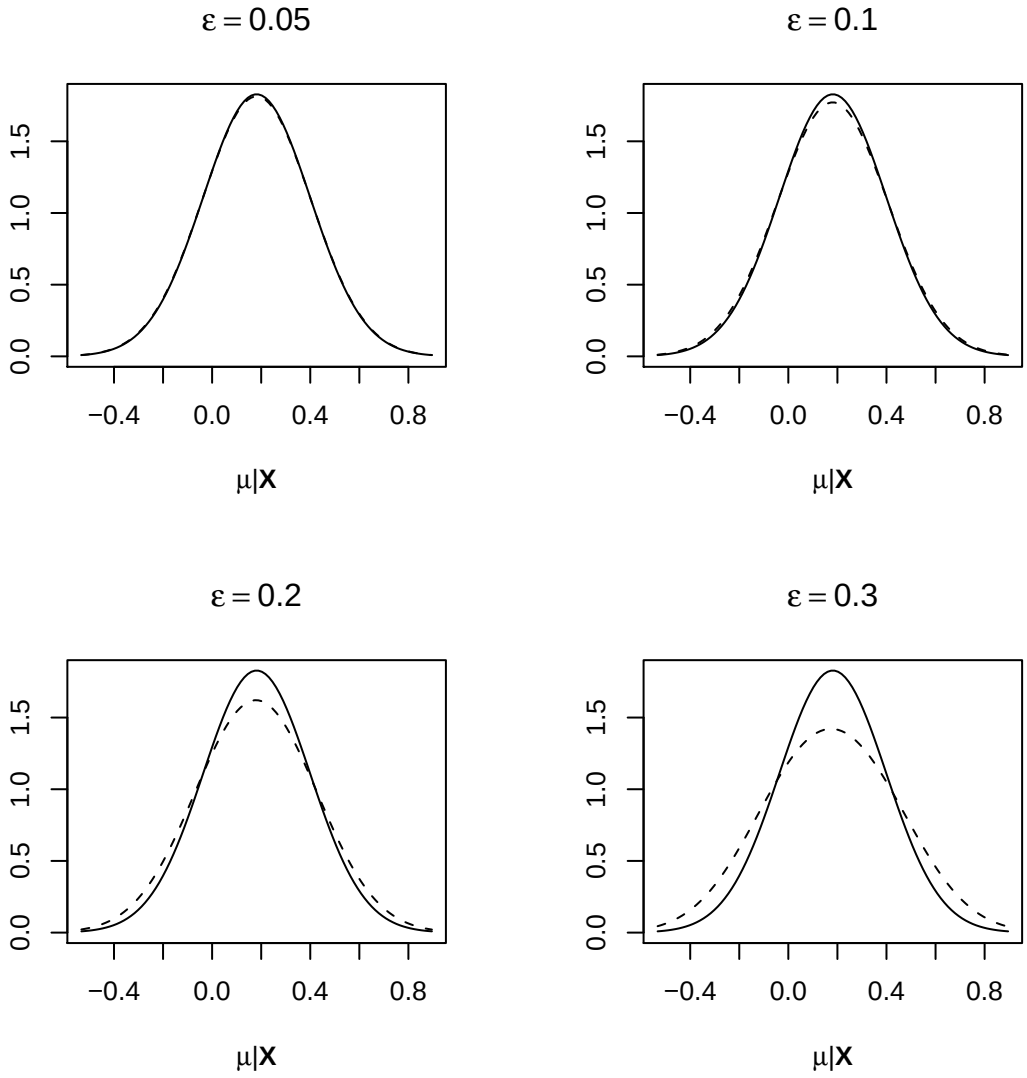


Abbildung 1.2: A-posteriori-Dichten aus Beispiel 1 (standardnormalverteilte Daten). Die unterbrochenen Linien stellen die ABC-aposteriori-Dichten dar, die durchgehenden Linien die wahre a-posteriori-Dichte.

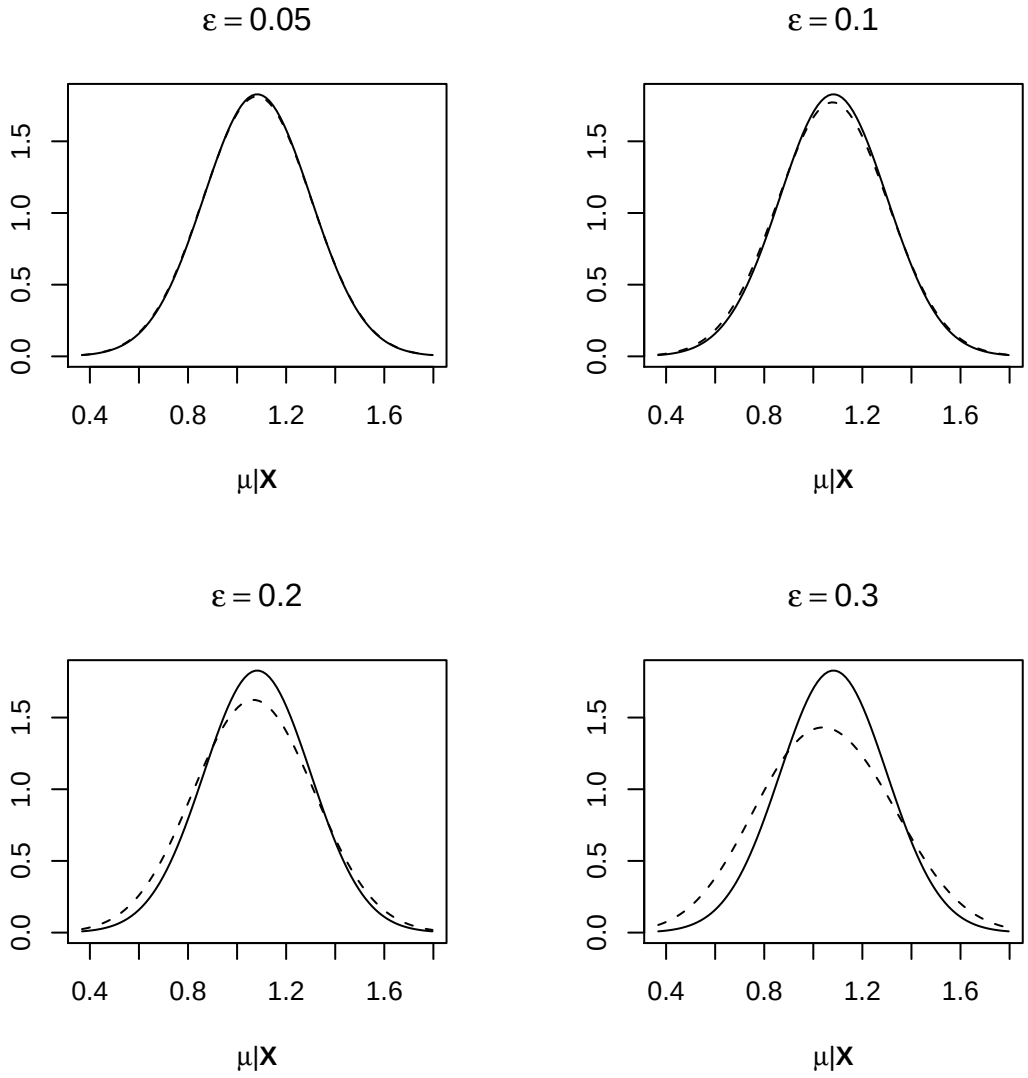


Abbildung 1.3: A-posteriori-Dichten aus Beispiel 2 (Verteilung der Daten: $X \sim \mathcal{N}(1,1)$). Die unterbrochenen Linien stellen die ABC-a-posteriori-Dichten dar, die durchgehenden Linien die wahre a-posteriori-Dichte.

Abbildung 1.3 zeigt, dass die ABC-a-posteriori-Dichte mit größerem ϵ nicht nur breiter wird, sondern sich auch vom Mittelwert der Daten ($\bar{X} = 1.14$) weg näher zum Erwartungswert der a-priori-Verteilung ($\mu_0 = 0$) hin verschiebt. Je größer ϵ ist, desto weniger Einfluss haben die Daten auf die ABC-a-posteriori-Verteilung und desto größer ist der Einfluss der a-priori-Verteilung.

Kapitel 2

Kreuzvalidierung

2.1 Einführende Definitionen

Es sei $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ mit $X_i \in \mathcal{X}$ für alle i ein Vektor von unabhängig identisch verteilten Zufallsvariablen, den Beobachtungen, mit gemeinsamer unbekannter Verteilung $\mathcal{F}$. Aus den Daten wird eine Eigenschaft $s \in \mathbb{S}$ von $\mathcal{F}$ geschätzt. Dabei kann es sich um den Parameter handeln, wenn $\mathcal{F}$ eine parametrische Verteilung ist, aber zum Beispiel auch um die Dichte. $\hat{s}$ ist ein Schätzer für s.

Definition 2.1. *Die Verlustfunktion eines Schätzers $\hat{s}$ für s ist*

$$\mathcal{L}(\hat{s}) = \mathcal{L}_{\mathcal{F}}(\hat{s}) := E_{X \sim \mathcal{F}}(\gamma(\hat{s}; X)), \tag{2.1}$$

wobei $\gamma : \mathbb{S} \times \Xi \mapsto [0, \infty)$ eine Kontrastfunktion ist und X eine einzelne Beobachtung (Arlot u. Celisse, 2010, S. 43).

Mit der Verlustfunktion wird die Qualität eines Schätzers gemessen. Eine häufig verwendete Kontrastfunktion ist der quadratische Kontrast $\gamma(\hat{s}, X) = (\hat{s} - X)^2$.

Definition 2.2. *Die Risikofunktion eines Schätzers $\hat{s}(x_1, \ldots, x_n)$ für s ist folgendermaßen definiert:*

$$E_{X_1, \ldots, X_n \sim \mathcal{F}} \left(l \left(s, \hat{s} \left(x_1, \ldots, x_n \right) \right) \right) \tag{2.2}$$

mit $l(s, \hat{s}) := \mathcal{L}_{\mathcal{F}}(\hat{s}) - \mathcal{L}_{\mathcal{F}}(s)$ (Arlot u. Celisse, 2010, S. 43).

Definition 2.3. *Ein statistischer Algorithmus $\mathcal{A}$ ist eine messbare Abbildung $\mathcal{A} : \bigcup_{n \in \mathbb{N}} \mathcal{X}^n \mapsto \mathbb{S}$ (Arlot u. Celisse, 2010, S. 44).*

Das heißt, jede Funktion, die einen Schätzer aus den Daten berechnet, ist ein statistischer Algorithmus (Arlot u. Celisse, 2010, S.42).

Der Output von $\mathcal{A}(\boldsymbol{X}) = \hat{s}^{\mathcal{A}}(\boldsymbol{X}) \in \mathbb{S}$ ist ein Schätzer für s. Die Qualität eines statistischen Algorithmus $\mathcal{A}$, gegeben Daten $\boldsymbol{X} \sim \mathcal{F}$ und eine fixe Kontrastfunktion γ, wird mit der Verlustfunktion $\mathcal{L}_{\mathcal{F}}(\hat{s}^{\mathcal{A}}(\boldsymbol{X}))$ gemessen, die so klein wie möglich sein sollte (Arlot u. Celisse, 2010, S. 44).

Definition 2.4. *Eine Teilmenge $S \subset \mathbb{S}$ ist ein Modell (Arlot u. Celisse, 2010, S. 44).*

Definition 2.5. *Ein Minimum-Kontrast-Schätzer über ein Modell S in $\mathbb{S}$ ist jeder Schätzer $\hat{s} \in \mathbb{S}$, der den empirischen Kontrast*

$$\mathcal{L}_{\mathcal{F}_n}(\hat{s}) = \frac{1}{n} \sum_{i=1}^{n} \gamma(\hat{s}, X_i) \tag{2.3}$$

minimiert, wobei $\mathcal{F}_n$ die empirische Verteilungsfunktion darstellt (Arlot u. Celisse, 2010, S. 44).

2.2 Modellselektion

In den meisten Fällen gibt es verschiedene plausible statistische Algorithmen, mit denen ein statistisches Problem gelöst werden kann. $(\hat{s}_\lambda)_{\lambda \in \Lambda}$ ist eine solche Familie von möglichen Algorithmen. Algorithmenselektion zielt darauf ab, einzig aus den Daten ein Modell $\hat{\lambda}(\boldsymbol{X}) \in \Lambda$ auszuwählen (Arlot u. Celisse, 2010, S. 45).

Modellselektion ist ein Spezialfall der Algorithmenselektion: Ein Algorithmus entspricht hier einem Modell und der Berechnung eines Minimum-Kontrast-Schätzers unter diesem Modell (Arlot u. Celisse, 2010, S. 45).

Es sei $(S_m)_{m \in \mathcal{M}}$ eine Familie von Modellen, das heißt $S_m \subset \mathbb{S}$. γ sei eine fixe Kontrastfunktion und für alle $m \in \mathcal{M}$ ist $\hat{s}_m(\boldsymbol{X})$ ein Minimum-Kontrast-Schätzer über S_m. Es soll ein Modell $\hat{m}(\boldsymbol{X}) \in \mathcal{M}$ mit Hilfe der Daten ausgewählt werden.

Modellselektion kann zwei verschiedene Ziele haben:

- Die Identifizierung des wahren Modells. In diesem Fall muss $\mathcal{M}$ das wahre Modell enthalten.
- Schätzung von s. Das wahre Modell muss nicht unbedingt in $\mathcal{M}$ enthalten sein. Es wird das Modell $\hat{m}$ ausgewählt, das die Risikofunktion von $\hat{s}_{\hat{m}}$ minimiert. Anschließend wird der Schätzer $\hat{s}_{\hat{m}}(\boldsymbol{X})$ berechnet.

Hier wird nur Modellselektion mit dem Ziel der Schätzung betrachtet.

Um dem Problem zu entgehen, dass dieselben Daten für die Auswahl von $\hat{\lambda}(\boldsymbol{X})$ und für die Berechnung von $\hat{s}_{\hat{\lambda}}(\boldsymbol{X})$ verwendet werden, was leicht zu „overfitting“ führen kann, kann Kreuzvalidierung verwendet werden.

2.3 Kreuzvalidierung

Kreuzvalidierung ist eine Technik zur Schätzung des Risikos eines statistischen Algorithmus $\mathcal{A}$.

Die Grundlage der Kreuzvalidierung ist die Validierung eines Modells, das an einen Datensatz angepasst wurde, durch neue Daten. In vielen Fällen ist es nicht möglich neue Daten zu erheben, daher wird der vorhandene Datensatz in ein Trainingssample zur Schätzung der Modellparameter und ein Validierungssample aufgeteilt. Damit soll „overfitting“ verhindert werden.

Es sei $I^{(t)} \subset \{1, \ldots, n\}$ und $I^{(v)} := (I^{(t)})^c \subset \{1, \ldots, n\}$, sodass $I^{(t)}$ und $I^{(v)}$ nicht leer sind.

$I^{(t)}$ ist die Indexmenge der Datenpunkte, die zum Trainingssample gehören, $\boldsymbol{X}^{(t)} := (X_i)_{i \in I^{(t)}}$ das Trainingssample und $n_t = \text{card}(I^{(t)})$ die Größe des Trainingssample. Die Notation für das Validierungssample ist analog mit Superskript (v).

Der einfachste Kreuzvalidierungsschätzer („hold out“) des Risikos des Algorithmus $\mathcal{A}(\boldsymbol{X})$ verwendet genau ein Trainings- und ein Validierungssample und ist folgendermaßen definiert:

$$\hat{\mathcal{L}}^{HO}\left(\mathcal{A}; \boldsymbol{X}; I^{(t)}\right) := \frac{1}{n_v} \sum_{i \in I^{(v)}} \gamma\left(\mathcal{A}\left(\boldsymbol{X}^{(t)}\right); X_i\right) \tag{2.4}$$

Damit kann eine allgemeine Definition für Kreuzvalidierungsschätzer formuliert werden:

Es sei $B \geq 1 \in \mathbb{N}$ und $I_1(t), \ldots, I_B(t)$ eine Folge von Trainingsindexmengen.

$$\hat{\mathcal{L}}^{CV}\left(\mathcal{A}; \boldsymbol{X}; \left(I_j^{(t)}\right)_{1 \leq j \leq B}\right) := \frac{1}{B} \sum_{j=1}^{B} \hat{\mathcal{L}}^{HO}\left(\mathcal{A}; \boldsymbol{X}; I_j^{(t)}\right) \tag{2.5}$$

Unterschiedliche Kreuzvalidierungsmethoden unterscheiden sich in der Auswahl und der Anzahl der Trainingssamples.

2.3.1 Verschiedene Kreuzvalidierungsmethoden

Leave-one-out

Jeder Datenpunkt wird einmal zur Schätzung ausgelassen und zur Validierung verwendet. Daher gilt: $n_t = n-1$, $B = n$ und $I_j^{(t)} = \{j\}^c$ für $j = 1, \ldots, n$.

$$\hat{\mathcal{L}}^{LOO}(\mathcal{A}, \boldsymbol{X}) = \frac{1}{n} \sum_{j=1}^{n} \gamma \left(\mathcal{A}(\boldsymbol{X}^{(-j)}), X_i \right), \tag{2.6}$$

wobei $\boldsymbol{X}^{(-j)} = (X_i)_{i \neq j}$ (Arlot u. Celisse, 2010, S. 54).

Leave-p-out

Jede mögliche Teilmenge der Größe p mit $p \in \{1, \ldots, n-1\}$ wird einmal zur Schätzung ausgelassen. Der LPO-Kreuzvalidierungsschätzer ist analog zum LOO-Kreuzvalidierungsschätzer definiert mit $B = \binom{n}{p}$. $(I_j^{(t)})_{1 \leq j \leq B}$ sind alle Teilmengen der Größe $n - p$ von $\{1, \ldots, n\}$ (Arlot u. Celisse, 2010, S. 54).

V-fache Kreuzvalidierung

Sowohl Leave-One-Out als auch Leave-p-out sind bei größeren Stichproben sehr aufwändig zu berechnen, da sie alle möglichen Teilmengen der Größe $n - p$ zur Validierung verwenden.

In der V-fachen Kreuzvalidierung mit $V \in \{1, \ldots, n\}$ werden die Daten zufällig in V etwa gleich große Subsamples unterteilt. Jedes Subsample wird einmal als Valdidierungsdatensatz verwendet.

$A_1, \ldots, A_V$ sind Partitionen von $\{1, \ldots, n\}$ mit $\text{card}(A_j) \approx n/V$ für alle j, $B = V$ und $I_j^{(t)} = A_j^c$ für $j = 1, \ldots, B$. Der V-fache Kreuzvalidierungsschätzer ist folgendermaßen definiert:

$$\hat{\mathcal{L}}^{VF}\left(\mathcal{A}, \boldsymbol{X}, (A_j)_{1 \leq j \leq V}\right) = \frac{1}{V} \sum_{j=1}^{V} \left(\frac{1}{\text{card}(A_j)} \sum_{i \in A_j} \gamma \left(\hat{s} \left(\boldsymbol{X}^{-(A_j)} \right), X_i \right) \right), \tag{2.7}$$

wobei $\boldsymbol{X}^{(-A_j)} = (X_i)_{i \in A_j^c}$ (Arlot u. Celisse, 2010, S. 54).

Weitere Kreuzvalidierungsschätzer und ähnliche Methoden werden zum Beispiel in (Arlot u. Celisse, 2010, ab S. 54) vorgestellt.

2.4 Eigenschaften von Kreuzvalidierungsschätzern

Es kann nicht allgemein bestimmt werden, wie gut ein Kreuzvalidierungsschätzer zur Modellselektion geeignet ist. Die Varianz und der Bias eines Kreuzvalidierungsschätzers für das Risiko eines Algorithmus können jedoch helfen, den Kreuzvalidierungsschätzer zu beurteilen. Die geringste Varianz und der kleinste Bias sind aber keine ausreichenden Kriterien für die beste Modellselektionsmethode (Arlot u. Celisse, 2010, S. 61). Da sich die Resultate nicht nur bezüglich der unterschiedlichen Kreuzvalidierungsmethoden unterscheiden, sondern auch bezüglich der statistischen Anwendung, werden hier nur die allgemeinsten Resultate zusammengefasst.

Kreuzvalidierung kann auch verwendet werden, wenn es sich bei $\hat{s}$ nicht um Minimum-Kontrast-Schätzer handelt, dann gelten aber die hier angeführten Resultate nicht.

2.4.1 Bias

Aus der Definition des Hold-Out-Schätzers 2.4 und aus der Tatsache, dass das Trainings- und das Validierungssample unabhängig sind, folgt:

$$E\left(\mathcal{L}^{\hat{HO}}\left(\mathcal{A}, \boldsymbol{X}, I^{(t)}\right)\right) = E\left(\frac{1}{n_v}\sum_{i\in I^{(v)}}\gamma\left(\mathcal{A}(\boldsymbol{X}^{(t)}); \xi_i\right)\right) \tag{2.8}$$

$$= \frac{1}{n_v}\sum_{i\in I^{(v)}} E\left(\gamma\left(\mathcal{A}(\boldsymbol{X}^{(t)}); \xi_i\right)\right) \tag{2.9}$$

$$= \frac{1}{n_v}\sum_{i\in I^{(v)}} \left(\mathcal{L}_P\left(\mathcal{A}\left(\boldsymbol{X}^{(t)}\right)\right)\right) \tag{2.10}$$

$$= E\left(\mathcal{L}_P\left(\mathcal{A}(\boldsymbol{X}^{(t)})\right)\right) \tag{2.11}$$

Daher gilt, nach der allgemeinen Definition für Kreuzvalidierungschätzer

(Gleichung 2.5), und wenn $\text{card}(I_j^{(t)}) = n_t$ für $j = 1, \ldots, B$:

$$E\left(\hat{\mathcal{L}}^{CV}\left(\mathcal{A}, \boldsymbol{X}, (I_j^{(t)})_{1\leq j\leq B}\right)\right) = E\left(\frac{1}{B}\sum_{j=1}^{B}\hat{\mathcal{L}}^{HO}(\mathcal{A}, \boldsymbol{X}, I_j^{(t)})\right) \tag{2.12}$$

$$= \frac{1}{B}\sum_{j=1}^{B} E\left(\hat{\mathcal{L}}^{HO}(\mathcal{A}, \boldsymbol{X}, I_j^{(t)})\right) \tag{2.13}$$

$$= \frac{1}{B}\sum_{j=1}^{B} E\left(\mathcal{L}_P(\mathcal{A}, \boldsymbol{X}^{(t)})\right) \tag{2.14}$$

$$= E\left(\mathcal{L}_P(\mathcal{A}, \boldsymbol{X}^{(t)})\right) \tag{2.15}$$

Da $\hat{\mathcal{L}}^{CV}\left(\mathcal{A}, \boldsymbol{X}, (I_k^{(t)})_{1\leq j\leq B}\right)$ ein Schätzer für das Risiko von $\mathcal{A}(\boldsymbol{X})$ ist, nämlich $E(\mathcal{L}_P(\mathcal{A}(\boldsymbol{X})))$, ist der Bias:

$$\text{Bias}\left(\hat{\mathcal{L}}^{CV}\left(\mathcal{A}, \boldsymbol{X}, (I_j^{(t)})_{1\leq j\leq B}\right)\right) = E(\mathcal{L}_P(\mathcal{A}(\boldsymbol{X}^{(t)}))) - E(\mathcal{L}_P(\mathcal{A}(\boldsymbol{X}))) \tag{2.16}$$

Der Bias gibt also die Differenz zwischen dem Risiko von $\mathcal{A}$ angewandt auf n_t und angewandt auf n Beobachtungen an. Da $n_t < n$ ist der Bias in den meisten Fällen negativ (außer in Fällen, in denen das Risiko nicht mit wachsendem n sinkt).

2.4.2 Varianz

Natürlich hängt auch die Varianz wesentlich von der Art des CV-Schätzers und der statistischen Fragestellung ab. Weitere Einflussfaktoren sind n_t und n_v sowie B. Allgemein gilt, dass die Varianz von CV-Methoden, die weniger als alle möglichen Kombinationen von Trainings- und Validierungssamples einer bestimmten Größe verwenden, höher ist, als die von LPO.

Unter der Annahme $\text{card}(I_j^{(t)}) = n_t \ \forall j$ gilt nach (Arlot u. Celisse, 2010, S. 59):

$$\text{var}\left(\hat{\mathcal{L}}^{HO}\left(\mathcal{A}; \boldsymbol{X}; I^{(t)}\right)\right) = \tag{2.17}$$

$$\frac{1}{n_v} E\left(var\left(\gamma(\hat{s}, \xi)|\hat{s} = \mathcal{A}(\boldsymbol{X}^{(t)})\right)\right) + var\left(\mathcal{L}_P(\mathcal{A}(\boldsymbol{X}^{(t)}))\right) \tag{2.18}$$

Der erste Term ist proportional zu $1/n_v$, also verringern mehr Daten zur Validierung die Varianz von $\hat{\mathcal{L}}^{HO}$. Weiters hängt die Varianz stark von der Verteilung von $\mathcal{L}_P(\mathcal{A}(\boldsymbol{X}^{(t)}))$ und daher auch davon, wie stark sich $\mathcal{A}$ bei einer kleinen Veränderung der Daten ändert.

2.5 Kreuzvalidierung in der Bayes-Statistik

Inferenz in der Bayes-Statistik basiert auf der a-posteriori-Verteilung der Parameter $\boldsymbol{\theta}$. Sie kann daher nicht in das in 2.1 beschriebene Konzept gefasst werden. Kreuzvalidierung kann aber dennoch auch in der Bayesianischen Statistik verwendet werden. Vorschläge dazu, wie Kreuzvalidierungsschätzer für numerisch aufwändige Bayesianische Verfahren konstruiert werden können, finden sich in Alqallaf u. Gustafson (2001). Eine mögliche Kontrastfunktion kann hier aus der Prognose-Verteilung abgeleitet werden.

Kapitel 3

Der Coalescent-Prozess

3.1 Einleitung

Der Coalescent-Prozess ist der wichtigste Prozess, mit dem Vererbungsprozesse in der Populationsgenetik modelliert werden.

Populationsgenetik beschäftigt sich damit, wie genetische Variabilität innerhalb einer Population zustande kommt. Ein wichtiger Faktor ist der Zufall (Gendrift): Welche Individuen sich mit welchen Individuen fortpflanzen und welche nicht, ist zum Teil zufällig. Andere Faktoren, deren Einfluss untersucht wird, sind Phänomene wie Mutation, Rekombination, Selektion und die demographische Struktur der Population (Wakeley, 2008, S. 1).

Eine Population besteht aus Individuen derselben Spezies, die sich, wenn es sich um diploide Individuen handelt, geographisch nahe genug beieinander aufhalten, um sich untereinander fortpflanzen zu können (Hartl u. Clark, 2007, S. 95).

3.2 Vom Wright-Fisher-Modell zum Coalescent-Prozess

Der Coalescent-Prozess beruht auf zwei grundlegenden Erkenntnissen: Erstens, dass es möglich (und vorteilhaft) ist, einen Vererbungsbaum ausgehend von einer Stichprobe aus der Population aus der Gegenwart in die Vergangenheit zu modellieren, und zweitens, dass unter der Annahme der neutralen Evolution (keine genetische Ausprägung bringt einen Selektionsvorteil) der Vererbungs- und der Mutationsprozess unabhängig voneinander sind und daher völlig getrennt betrachtet werden können (Nordborg, 2007, S. 2ff). Mathematisch exakt wurde der Coalescent-Prozess erstmals von Kingman (1982)

beschrieben.

3.2.1 Das Wright-Fisher-Modell

Vorläufig werden nur haploide Individuen betrachtet, das heißt, es gibt nur ein Geschlecht. Ein Individuum pflanzt sich fort, indem es die eigenen Gene weiter gibt. Wie die Fortpflanzung von diploiden Individuen bzw. zwei Geschlechtern modelliert werden kann, wird im Kapitel 3.4.2 beschrieben.

Es sei N die Größe der Population, $g = 0$ ist die gegenwärtige Generation, aus der eine Stichprobe von n Individuen gezogen wird. Es werden DNA-Sequenzen dieser n Individuen betrachtet (um diploide Individuen leichter berücksichtigen zu können, wird in der Literatur oft $2N$ anstatt N als Populationsgröße verwendet). Die Generationen werden von der Gegenwart in die Vergangenheit gezählt, d. h. die Generation g „lebte“ vor g Generationen.

Dem Wright-Fisher-Modell liegen folgende Annahmen zugrunde (Hein u. a., 2005, S. 13):

1. Diskrete und nicht überlappende Generationen
2. Haploide Individuen
3. Konstante Populationsgröße
4. Neutrale Evolution
5. Keine geographische oder soziale Struktur
6. Keine Rekombination

Ein Individuum der Generation g entsteht, indem ein Individuum der Generation $g + 1$ zufällig ausgewählt und seine DNA kopiert wird. Das wird so oft wiederholt, bis die Generation g aus N Individuen besteht. Daher hat jedes Individuum genau einen Vorfahren, aber nicht jedes pflanzt sich fort.

3.2.2 Herleitung des Coalescent-Prozesses

Der Coalescent-Prozess basiert auf der wichtigen Erkenntnis, dass es möglich ist, von der letzten Generation aus Erkenntnisse über die Vergangenheit, also den Prozess, der zur Entstehung dieser Generation geführt hat, zu gewinnen (Nordborg, 2007, S. 3). Das Wright-Fisher-Modell wird also „rückwärts“ angewandt. Das heißt, man kann sich vorstellen, dass sich jedes der N Individuen der Generation g seinen Vorfahren in der Generation $g+1$ zufällig „wählt“.

Wenn zwei oder mehrere Individuen dasselbe Individuum auswählen, verbinden sich ihre Vererbungslinien in einem gemeinsamen Vorfahren - englisch: to coalesce. Das Ereignis von Interesse ist also bei rückwärtiger Betrachtung das Verschmelzen der Linien.

Der Coalescent-Prozess ist der Grenzprozess, gegen den das Wright-Fisher-Modell konvergiert, wenn die Zeit als kontinuierlich betrachtet wird und $N \to \infty$. Wie das im Detail funktioniert, wird im Folgenden beschrieben.

Es sei T_k die Anzahl der Generationen (bzw. im kontinuierlichen Modell, das später eingeführt wird, die Zeit auf der kontinuierlichen Skala), während der $k \leq n$ Individuen aus der Stichprobe genau k verschiedene Vorfahren haben.

Die Wahrscheinlichkeit, dass zwei Individuen in der vorherigen Generation zwei verschiedene Vorfahren haben, ist

$$1 - \frac{1}{N}, \tag{3.1}$$

da die Wahrscheinlichkeit, dass zwei Individuen vom selben Individuum abstammen, $1 \cdot \frac{1}{N}$ beträgt.

Daher, und weil die zufällige Wahl eines Vorfahren in jeder Generation unabhängig von den anderen Generationen ist, ist die Wahrscheinlichkeit, dass zwei Individuen genau g Generationen zurück einen gemeinsamen Vorfahren haben,

$$\left(1 - \frac{1}{N}\right)^{g-1} \frac{1}{N} \tag{3.2}$$

T_2 ist also folgendermaßen verteilt:

$$P(T_2 = g) = \left(1 - \frac{1}{N}\right)^{g-1} \frac{1}{N} \text{ für } g = 1, 2, \ldots \tag{3.3}$$

Das entspricht einer geometrischen Verteilung.

Es gilt daher

$$T_2 \sim Geom\left(\frac{1}{N}\right), \tag{3.4}$$

und weiters

$$E(T_2) = N. \tag{3.5}$$

Die Wahrscheinlichkeit, dass $k \leq n$ Individuen jeweils verschiedene Vorfahren in der vorigen Generation haben, ist

$$\begin{aligned} P(&\text{Individuum 1 hat andere Vorfahren als die Individuen } 2, \ldots, k \wedge \\ &\wedge \text{ Individuum 2 hat andere Vorfahren als die Individuen } 3, \ldots, k \wedge \\ &\wedge \ldots \wedge \text{ Individuum } k-1 \text{ hat andere Vorfahren als das Individuum} k) \end{aligned}$$

$$= P(\text{Individuum 1 hat andere Vorfahren als die Individuen } 2, \ldots, k) \cdot$$
$$\cdot P(\text{Individuum 2 hat andere Vorfahren als die Individuen } 3, \ldots, k) \cdot$$
$$\cdot \ldots \cdot P(\text{ Individuum } k-1 \text{ hat andere Vorfahren als das Individuum } k)$$

$$= \prod_{i=1}^{k-1} (1 - \frac{i}{N}) \tag{3.6}$$

$$= 1 - \sum_{i=1}^{k-1} \frac{i}{N} + O(\frac{1}{N^2}) \tag{3.7}$$

$$= 1 - \frac{\binom{k}{2}}{N} + O(\frac{1}{N^2}) \tag{3.8}$$

Wenn $N \to \infty$, kann $O(\frac{1}{N^2})$ vernachlässigt werden. Gleichzeitig heißt das, dass in jeder Generation höchstens ein Paar von Individuen einen gemeinsamen Vorfahren findet (Hein u. a., 2005, S. 22): Die Wahrscheinlichkeit, dass zwei Individuen einen gemeinsamen Vorfahren haben, ist $\binom{k}{2}/N$, die Wahrscheinlichkeit, dass mehr als zwei Individuen gemeinsame Vorfahren haben, ist $O(\frac{1}{N^2})$.

Daraus kann die Verteilung von T_k abgeleitet werden:

$$P(T_k = g) = \left(1 - \frac{\binom{k}{2}}{N}\right)^{g-1} \frac{\binom{k}{2}}{N} \tag{3.9}$$

Es gilt also:

$$T_k \sim Geom\left(\frac{\binom{k}{2}}{N}\right) \tag{3.10}$$

Um von den diskreten Generationen zu einer kontiniuerlichen Zeitskala zu gelangen, wird die diskrete Skala wird $E(T_2) = N$, der erwarteten Zeit, bis zwei genealogische Linien sich treffen, skaliert. Es sei t die kontinuierliche Zeit und g die Anzahl der Generationen:

$$t = \frac{g}{N} \tag{3.11}$$

Eine Einheit der kontinuierlichen Zeit entspricht also N. Durch diese Skalierung wird der resultierende Coalescent-Prozess unabhängig von N, das heißt, die Form des Vererbungsbaumes wird unabhängig von N (Hein u. a., 2005, S. 24).

Da $p = \binom{k}{2}/N$, der Parameter der geometrischen Verteilung, bei wachsendem N klein wird und die Zeit kontinuierlich ist, konvergiert die geometrische Verteilung gegen eine Exponentialverteilung (Hein u. a., 2005, S. 24) und es gilt:

$$T_k \sim Exp\left(\binom{k}{2}\right) \tag{3.12}$$

3.2.3 Der Coalescent-Prozess als Markov-Prozess

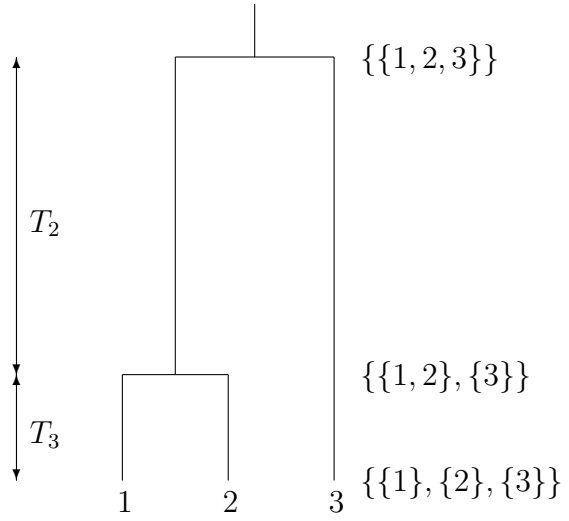
In einer geeigneten Formulierung wird erkennbar, dass der Coalescent-Prozess ein Markov-Prozess auf einem diskreten Zustandsraum ist (Nordborg, 2007, S. 6): Es sei $\mathcal{E}_n$ der Zustandsraum, der aus allen Äquivalenzrelationen auf $\{1, \ldots, n\}$ besteht, und $Q = (q_{i,j})_{i,j \in \mathcal{E}_n}$ die Intensitätsmatrix mit

$$q_{i,j} = \begin{cases} -k(k-1)/2 & i = j \\ 1 & i \prec j \\ 0 & \text{sonst} \end{cases} \tag{3.13}$$

wobei $k = |i|$ die Anzahl der Äquivalenzklassen in i ist. $i \prec j$ gilt genau dann, wenn j aus i durch das Verschmelzen von zwei Äquivalenzklassen von i entsteht.

Die Äquivalenzrelation hat die Bedeutung „hat (zu diesem Zeitpunkt) gemeinsame Vorfahren“.

Anhand eines Beispiels kann diese abstrakte Darstellung illustriert werden:



Der Zustandsraum $\mathcal{E}$ besteht aus folgenden Zuständen:

$$\alpha = \{\{1\},\{2\},\{3\}\} \tag{3.14}$$
$$\beta_1 = \{\{1,2\},\{3\}\} \tag{3.15}$$
$$\beta_2 = \{\{1\},\{2,3\}\} \tag{3.16}$$
$$\beta_3 = \{\{1,3\},\{2\}\} \tag{3.17}$$
$$\gamma = \{\{1,2,3\}\} \tag{3.18}$$

Die Intensitätsmatrix Q ist also, nach Gleichung 3.13:

Q	α	β_1	β_2	β_3	γ
α	$\frac{-3(3-1)}{2} = -3$	1	1	1	0
β_1	0	$\frac{-2(2-1)}{2} = -1$	0	0	1
β_2	0	0	$\frac{-2(2-1)}{2} = -1$	0	1
β_3	0	0	0	$\frac{-2(2-1)}{2} = -1$	1
γ	0	0	0	0	$\frac{-1(1-1)}{2} = 0$

Die positiven Einträge in Q geben den Parameter der Exponentialverteilung an: Die Zeit, bis zwei Individuen einen gemeinsamen Vorfahren „finden“, T_2, ist (laut Gleichung 3.10) exponentialverteilt mit Parameter 1. Vom Zustand α ausgehend gibt es drei gleich wahrscheinliche Möglichkeiten, dass sich Linien zweier Individuen treffen bzw. dass zwei Äquivalenzklassen verschmelzen, nämlich den Übergang zu β_1, β_2 oder β_3. Die Einträge Q_{α,β_i} sind daher 1 für alle $i = 1, 2, 3$.

Einträge, die 0 oder negativ sind, bedeuten, dass dieser Übergang nicht möglich ist. In der letzten Zeile von Q sind alle Einträge 0, das heißt, γ ist eine absorbierende Klasse. Die Einträge in der Diagonale sind negativ, weil ein Übergang in den selben Zustand nicht möglich ist und die Zeilensumme jeweils 0 sein muss.

3.2.4 Mutation

Das bisher beschriebene Modell beschreibt nur den Zufall in der Fortpflanzung (Gendrift): Einige Individuen pflanzen sich fort, andere nicht. Oder, rückwärts in der Zeit betrachtet: Die Verteilung der Zeit, die es dauert, bis sich zwei Vererbungslinien zufällig treffen.

Unabhängig davon wird im Coalescent-Prozess die genetische Variabilität der Individuen mittels „gene dropping“ modelliert: Die Allelen der ersten Generation werden in passender Weise festgelegt und in jeder folgenden Generation an die Nachkommen weitergegeben, außer, wenn es zu einer Mutation

kommt. Wenn nur eine Stichprobe der Population betrachtet wird, genügt es, dem letzten gemeinsamen Vorfahren (Most Recent Common Ancestor – MRCA) ein Allel zuzuordnen (Nordborg, 2007, S. 3).

Aufgrund der Annahme der neutralen Evolution werden Mutationen unabhängig von der Baumstruktur modelliert.

Es sei μ die Mutationsrate pro Individuum pro Generation. Um vom diskreten Wright-Fisher-Modell zum kontinuierlichen Coalescent-Prozess zu gelangen, wird die *skalierte Mutationsrate* $\theta = 2N\mu$ betrachtet.

Mutationen ereignen sich auf einem „Ast“ des Baumes nach einem Poisson-Prozess mit Rate $\theta/2$. Das heißt, die Anzahl der Mutationen auf einem Ast der Länge τ ist Poisson-verteilt mit Erwartungswert $\tau\theta/2$ und die Mutationen sind über die Länge des Astes gleichverteilt (Nordborg, 2007, S. 27).

3.2.5 Weitere statistische Größen

Gesamtlänge der Äste eines Baumes

Die Gesamtlänge der Äste eines Baumes, vom MRCA bis zu den n Individuen in der Stichprobe der Generation 0, ist folgendermaßen definiert (Tavaré u. a., 1997, S. 506):

$$L_n = \sum_{j=2}^{n} jT_j \tag{3.19}$$

mit Erwartungswert

$$E(L_n) = \sum_{j=2}^{n} jE(T_j) = 2\sum_{j=1}^{n-1} \frac{1}{j} \tag{3.20}$$

Anzahl der Mutationen

Unter dem Infinite-Sites-Modell, das heißt, unter der Annahme, dass jede Mutation an einer neuen Position der DNA-Sequenz passiert und es an keiner Position mehr als eine Mutation geben kann, entspricht die Anzahl der Mutationen auf allen Ästen des Baumes der Anzahl der Positionen mit unterschiedlichen Ausprägungen auf der DNA-Sequenz in der Stichprobe (engl: „segregating sites“).

Wenn die Gesamtlänge der Äste $L_n = l$ gegeben ist, ist die Anzahl der segregating sites S_n Poisson-verteilt:

$$S_n|L_n = l \sim Poisson(\frac{\theta l}{2}), \tag{3.21}$$

wobei θ die skalierte Mutationsrate ist, da die Zeitdauern zwischen zwei Mutationen exponentialverteilt sind (siehe Kapitel 3.2.4) (Tavaré u. a., 1997, S. 507).

Es kann gezeigt werden, dass gilt (Hein u. a., 2005, S. 60):

$$E(S_n) = \theta \sum_{j=1}^{n-1} \frac{1}{j} \tag{3.22}$$

Die Wahrscheinlichkeitsfunktion ist laut Hein u. a. (2005, S. 59):

$$s_n(k) := P(S_n = k) = \frac{n-1}{\theta} \sum_{i=1}^{n-1} (-1)^{i-1} \binom{n-2}{i-1} \left(\frac{\theta}{i+\theta}\right)^{k+1} \tag{3.23}$$

Die Verteilungsfunktion wird mit $\mathcal{S}_n$ bezeichnet.

3.3 Allgemeinheit des Coalescent-Prozess

Das Wright-Fisher-Modell ist nur eines von vielen neutralen Modellen, das mit $N \to \infty$ gegen den Coalescent-Prozess konvergiert. Auch viele biologische Phänomene, die ein komplexeres Modell als das Wright-Fisher-Modell benötigen, wie zum Beispiel Migration oder zwei separate Geschlechter, können mit dem Coalescent-Prozess abgebildet werden, genauso wie andere Modelle, wie beispielsweise das Moran-Modell, das überlappende Generationen verwendet (Nordborg 2007, S. 8, Hein u. a. 2005, S. 31).

In vielen Fällen genügt es, die lineare Skalierung der Zeit, die im Wright-Fisher-Modell mit N erfolgt, zu verändern. In einem Modell, das einen anderen Skalierungsfaktor benötigt, wird mit der sogenannten *effektiven Populationsgröße* N_e skaliert, die neben der Populationsgröße N vom Modell abhängt (Nordborg, 2007, S. 9).

Einige Erweiterungen des Wright-Fisher-Modells, die in den Coalescent integriert werden können, werden im nächsten Kapitel diskutiert.

3.4 Erweiterungen des Modells

3.4.1 Nicht-konstante Populationsgröße

Es sei $N(t)$ die Populationsgröße in der Generation t. Wenn $N(t)$ deterministisch und für alle t bekannt ist, kann die Veränderung der Populationsgröße in den Coalescent-Prozess aufgenommen werden. Hier genügt allerdings eine lineare Skalierung mit der effektiven Populationsgröße N_e nicht, da diese nicht konstant über die Generationen ist.

Die Zeit wird nicht-linear skaliert, wobei „seit g Generationen“ bedeutet, „seit

$$s(g) = \sum_{i=1}^{g} \frac{1}{N(i)} \tag{3.24}$$

Einheiten“ Coalescence-Zeit.

Damit ein solcher Prozess gegen den Coalescent-Prozess konvergiert, muss außerdem sicher gestellt sein, dass $N(t)$ in jeder Generation groß ist (Nordborg, 2007, p. 10).

3.4.2 Subpopulationen

Auch die Vererbungsmechanismen von Populationen, die in Subpopulationen unterteilt sind, zwischen denen eine bestimmte Art von Austausch von Genen oder Individuen herrscht, können mit dem Coalescent-Prozess modelliert werden. Beispiele dafür sind die geographische Segregation, bei der örtlich getrennte Gruppen Individuen mit einer bestimmten Migrationsrate austauschen, aber auch die Fortpflanzung diploider Individuen.

Nordborg (2007, S. 11) beschreibt ein „strukturiertes Wright-Fisher-Modell“, das gegen den Coalescent konvergiert: Die Populationsgröße N ist fix, die Größen der M Subpopulationen, $N_i, i = 1, \ldots, M$ ebenso. Jedes Individuum hat eine unendliche Anzahl von Nachkommen, die mit der Wahrscheinlichkeit $m_{i,j}$ von der Subpopulation i in die Subpopulation j wandern. Die nächste Generation von Individuen entsteht dadurch, dass aus den Nachkommen zufällig gezogen wird, bis jede Subpopulation ihre Größe N_i erreicht hat.

Geographische Segregation

Unter bestimmten Annahmen konvergiert das „strukturierte Wright-Fisher-Modell“ gegen den „strukturierten Coalescent“.

Im Grenzwert gibt es nunmehr zwei statt einem möglichen Ereignis:

- Zwei Individuen „finden“ einen gemeinsamen Vorfahren innerhalb einer Gruppe (Coalescent)
- Ein Individuum wechselt von einer Gruppe in eine andere (Migration)

Diese folgen zwei unabhängigen Poisson-Prozessen.

Die resultierenden Coalescent-Bäume unterscheiden sich vom Standard-Coalescent nicht nur durch die Länge der Äste und damit die Zeitskalierung, sondern auch in ihrer Topologie. Wenn die Migrationsraten gering sind, finden

zwei Individuen derselben Subpopulation schnell einen gemeinsamen Vorfahren, während es lange dauern kann, bis die Vorfahren von zwei Individuen verschiedener Subpopulationen zusammentreffen.

Diploide Individuen

Geschlechtliche Fortpflanzung bzw. diploide Individuen können modelliert werden, indem die Populationsgröße (und damit der Skalierungsfaktor) $2N$ anstatt N ist und Subpopulationen gebildet werden.

Im Fall von Hermaphroditen, also Individuen, die sich entweder selbst oder mit einem anderen Individuum fortpflanzen, werden N Gruppen der Größe 2 gebildet. Bei Individuen, die entweder männlich oder weiblich sind und sich daher ausschließlich geschlechtlich fortpflanzen, werden die Geschlechter als zwei Subpopulationen betrachtet. Wenn die Population aus N_m männlichen und N_f weiblichen Individuen besteht, wobei $N = N_m + N_f$, kann die geschlechtliche Fortpflanzung als die einer haploiden Population der Größe $2N$, die in zwei Subpopulationen der Größe $2N_f$ beziehungsweise $2N_m$ aufgeteilt sind, die wieder in Untergruppen der Größe 2 geteilt sind, modelliert werden. In beiden Fällen ist es möglich, die effektive Populationsgröße N_e zu berechnen und mit einer linearen Skalierung der Zeit zum Standard-Coalescent zu gelangen.

3.4.3 Rekombination

Rekombination bedeutet, dass ein Chromosom nicht als Ganzes (mit oder ohne Mutation) von einem Vorfahren übernommen wird, sondern dass das Chromosom an einem bestimmten Punkt in zwei Teile geteilt wird. Diese stammen von unterschiedlichen Individuen, in einer diploiden Population von den beiden Elternteilen.

Ins Wright-Fisher-Modell kann Rekombination folgendermaßen integriert werden: Eine DNA-Sequenz sucht sich aus der vorigen Generation zwei DNA-Sequenzen aus – nicht nur eine, wie im Standard-Modell. Mit einer bestimmten Wahrscheinlichkeit rekombinieren sich diese an einem zufälligen Punkt auf der Sequenz; wenn nicht, wird eine der Sequenzen einfach kopiert (Hein u. a., 2005, S. 138).

Greift man einen einzelnen Punkt oder ein Intervall, auf dem keine Rekombination stattgefunden hat, aus der Sequenz heraus, so entspricht die Baumstruktur, die zu seiner Entstehung geführt hat, dem Modell ohne Rekombination (Hein u. a., 2005, S. 138).

Betrachtet man den Fortpflanzungsprozess rückwärts in der Zeit, können zwei Ereignisse beobachtet werden:

- Zwei genealogische Linien treffen sich (Coalescent)
- Eine genealogische Linie teilt sich (Rekombination)

Diese werden mit unabhängigen Poisson-Prozessen modelliert (Nordborg, 2007, S. 18).

3.4.4 Selektion

Neutrale Evolution ist ein essenzieller Punkt in der Modellierung des Coalescent: Nur unter der Annahme, dass es keine Mutationen gibt, die einen Selektionsvorteil oder -nachteil mit sich bringen, können der Coalescent-Baum und der Mutationsprozess unabhängig voneinander modelliert werden.

Nichtsdestotrotz gibt es Modelle, die den Coalescent mit Selektion verbinden können. Zwei Methoden werden in Nordborg (2007, ab S. 22) kurz beschrieben:

- „The ancestral selection graph“: Der Vererbungsbaum wird, wie im Standard-Coalescent, von der Gegenwart aus in die Vergangenheit konstruiert. Anschließend werden Mutationen mittels „gene-dropping“ produziert. Abschließend wird der Baum „gestutzt“: Äste mit Allelen, die einen selektiven Nachteil bringen, werden entfernt.

- „The conditional structured coalescent“: Die Population wird in Allelen-Klassen aufgeteilt, innerhalb derer es keine Selektion gibt. Wenn die Größen dieser Klassen bekannt sind, kann die im Kapitel 3.4.2 beschriebene Methodik für Subpopulationen verwendet werden. Der Wechsel von Individuen zwischen Klassen, der geographisch als Migration angesehen wird, entspricht hier der Mutation zu einem mehr oder weniger vorteilhaften Allel.

3.5 Likelihood

Es sei N die Populationsgröße und μ die Mutationsrate pro Individuum pro Generation. $\boldsymbol{X} = (X_1, \ldots, X_n)$ sind die erhobenen Daten, also die Gen-Sequenzen von n zufällig gezogenen Individuen der Generation 0.

Wenn man den Coalescent-Prozess ohne Erweiterungen betrachtet, sind N und μ die einzigen Parameter.

Die Likelihood von N und μ

$$L(N, \mu) = P(\boldsymbol{X}|N, \mu) \tag{3.25}$$

kann für N und μ nicht separat ausgewertet werden, da sie vom Produkt von N und μ abhängt (Stephens, 2007, S. 880). Daher wird der skalierte Mutationsparameter $\theta = 2N\mu$ betrachtet:

$$L(\theta) = P(\boldsymbol{X}|\theta) \tag{3.26}$$

Wenn die Baumstruktur T bekannt ist, kann

$$P(\boldsymbol{X}|T, \theta) \tag{3.27}$$

berechnet werden (Stephens, 2007, S. 884). Üblicherweise sind aber die Daten $\boldsymbol{X}$ die einzige Information, daher müssen alle möglichen Baum-Topologien in Betracht gezogen werden und es kann leicht gezeigt werden, dass

$$L(\theta) = \sum_{T \in \mathcal{T}} P(\boldsymbol{X}|T, \theta) P(T|\theta), \tag{3.28}$$

wobei $\mathcal{T}$ die Menge aller möglichen Bäume ist.

Alle Summanden können berechnet werden, allerdings ist die Anzahl der möglichen Baumtopologien so groß, dass die Berechnung der Summe sehr viel Zeit in Anspruch nimmt. Dazu kommt, dass die Länge jeden Astes von 0 bis ∞ variieren kann, und daher zusätzlich für jeden möglichen Baum Integrale der Dimension $N - 1$ berechnet werden müssen (Felsenstein, 2006, S. 692).

Um diese aufwändigen bzw. für schon für relativ kleine Stichprobengrößen unmöglichen Berechnungen zu umgehen, müssen entweder andere Schätzer als Maximum-Likelihood-Schätzer oder Simulationsmethoden verwendet werden. Das gilt nicht nur für die hier exemplarisch beschriebene skalierte Mutationsrate θ, sondern auch für andere Parameter im erweiterten Coalescent-Prozess.

3.6 Watterson-Schätzer für θ

Watterson-Schätzer:

$$\hat{\theta}_W = \frac{S_n}{\sum_{j=1}^{n-1} 1/j} \tag{3.29}$$

wobei S_n die beobachtete Anzahl der „segregating sites“ in der Stichprobe ist. Der Watterson-Schätzer ist ein Momentenschätzer und wird aus Gleichung 3.22 hergeleitet. Für konstante Populationsgröße und eine Population, die nicht in Subpopulationen unterteilt ist, ist er erwartungstreu.

Teil II

Simulation

Kapitel 4

Beschreibung

4.1 Einleitung

Wie gut die a-posteriori-Verteilung von einer mittels ABC simulierten Verteilung angenähert wird, hängt unter anderem von der Wahl von ϵ im Rejection-Algorithmus ab. Wie schon im Kapitel 1.3.3 beschrieben, gilt: Je kleiner ϵ, desto näher ist die simulierte a-posteriori-Verteilung der wahren Verteilung. Gleichzeitig nimmt aber bei einem kleineren ϵ die Wahrscheinlichkeit zu, dass ein simuliertes $\boldsymbol{\theta}^*$ verworfen wird und die Anzahl der $\boldsymbol{\theta}^*$'s, aus denen die a-posteriori-Dichte geschätzt wird, nimmt ab. Diese Schätzung ist dadurch stärkeren Zufallsschankungen ausgesetzt. Die Anzahl der Wiederholungen des Rejection-Algorithmus, m, kann jedoch nicht beliebig erhöht werden, da die benötigte Rechenzeit, je nach Komplexität des Modells, Grenzen setzt.

4.1.1 Beispiel: Standard-Coalescent-Prozess

Das Modell $\mathcal{M}$ sei ein Standard-Coalescent-Prozess, der nur von dem eindimensionalen Parameter $\theta = 2N\mu$, der skalierten Mutationsrate, abhängt (siehe Abschnitt 3.2.4). Außerdem wird das Infinite-Sites-Modell angenommen (siehe Abschnitt 3.2.5). Als a-priori-Dichte $\pi(\theta)$ wird eine Gleichverteilung auf dem Intervall $(0, 10)$ angenommen. Die a-posteriori-Dichte kann, laut Abschnitt 3.5, nicht analytisch bestimmt werden und wird daher mit ABC simuliert.

Der ABC-Algorithmus sieht folgendermaßen aus:

Algorithmus 4.1. *θ hat die a-priori-Verteilung $\pi(\theta)$. Ein Coalescent-Prozess mit dem Parameter θ hat in der letzten Generation n Nachkommen. Die Daten dieser n Individuen werden mit $\boldsymbol{X}$ bezeichnet.*

1. *Aus $\boldsymbol{X}$ wird S, die Anzahl der Mutationen, berechnet.*

2. *Rejection-Algorithmus: m-malige Wiederholung der Schritte*

   (a) *Ziehe* θ^* *aus* $\pi(\theta)$.

   (b) *Simuliere* S^* *aus dem Coalescent-Prozess mit Parameter* θ^*.

   (c) *Verwirf* S^*, *wenn* $d(S, S^*) = |S - S^*| > \epsilon$, *sonst speichere* θ^* *als* θ_i^*, *wenn* θ^* *der i-te nicht verworfene simulierte Wert ist.*

3. *Aus* $(\theta_i^*)_{1 \leq i \leq w}$, *wobei w die Anzahl der nicht verworfenen Werte ist, wird der Schätzer* $\hat{p}_{ABC}(\theta|S)$ *für die a-posteriori-Dichte* $p(\theta|\boldsymbol{X})$ *mit Hilfe eines Kerndichteschätzers berechnet.*

Details zur Implementierung in R und MS können Abschnitt 4.2.1 entnommen werden.

Um die Auswirkungen der Wahl von ϵ auf die simulierte a-posteriori-Dichte darzustellen, wird der ABC-Algorithmus mit $m = 10000$ für verschiedene Werte von ϵ durchgeführt. Mittels MS wird der Datensatz 1 mit $\theta = 5$ und $n = 50$ gezogen. Der simulierte Datensatz enthält 21 Mutationen. Der daraus berechnete Watterson-Schätzer ist $\hat{\theta}_W = 4.69$.

Die Anzahlen der θ^*, die nicht verworfen werden, sind in Tabelle 4.1 angeführt.

ϵ	Anzahl
0	228
1	678
2	1112
3	1560
5	2515
10	4709

Tabelle 4.1: Anzahl der akzeptierten θ^* in Datensatz 1

Aus den akzeptierten θ^* werden die a-posteriori-Dichten jeweils mit einem Kerndichteschätzer mit Normalverteilungskern geschätzt. Für alle ϵ sind so viele Werte vorhanden, dass eine Kerndichteschätzung sinnvoll erscheint (mindestens 228). Die Ergebnisse sind in Abbildung 4.1 dargestellt.

Es ist eindeutig erkennbar, dass sich die simulierten a-posteriori-Dichten mit wachsendem ϵ der Form der a-priori-Dichte nähern. Die Form der Dichten hängt auch stark vom gewählten Kern ab: Vor allem an den Enden, wo der Träger der Dichte über die tatsächlich simulierten Werte hinausgeht, ist die Form der Normalverteilung gut sichtbar.

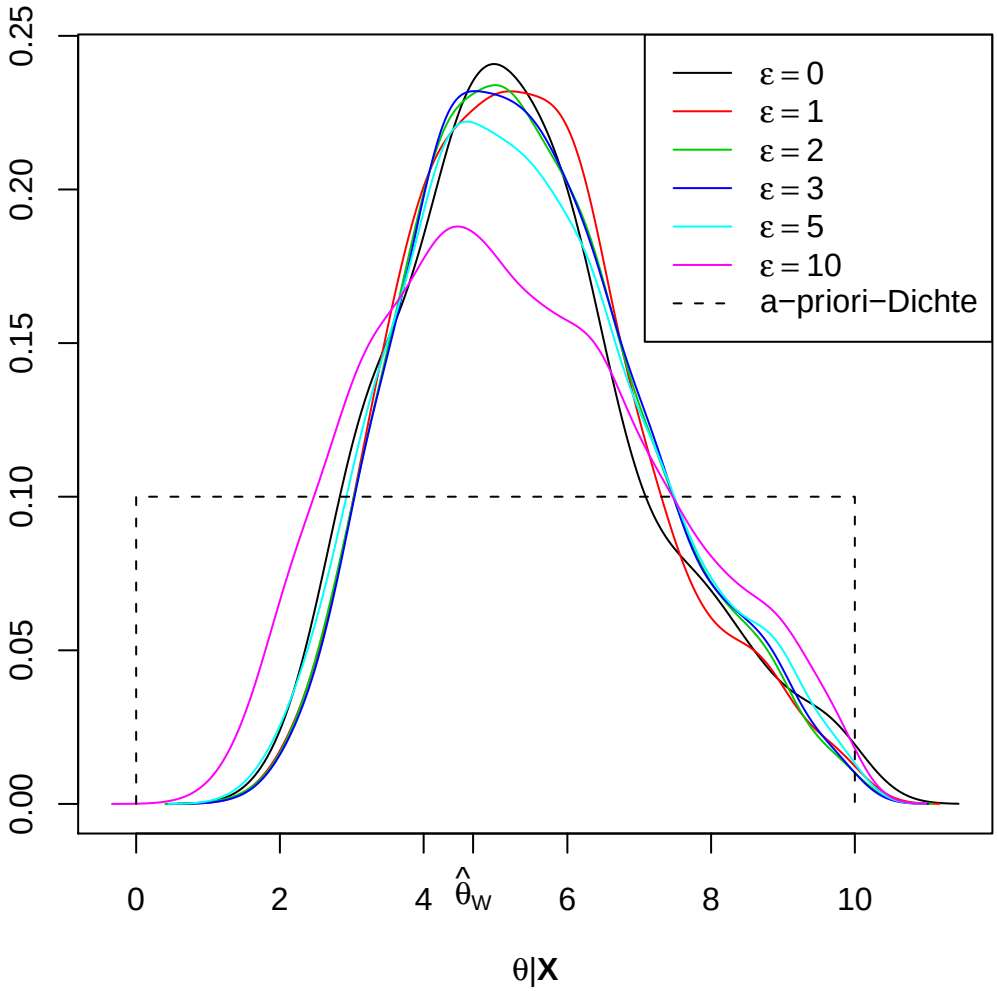


Abbildung 4.1: Mittels ABC simulierte a-posteriori-Dichten von θ in Datensatz 1 bei verschiedenen Werten für ϵ.

Mit kleiner werdendem ϵ nähern sich die simulierten Dichten der wahren a-posteriori-Dichte an. Wie gut diese Annäherung ist, kann nicht festgestellt werden, da die wahre a-posteriori-Dichte nicht bekannt ist.

Zum Vergleich wird Datensatz 2 mit $\theta = 1$ simuliert. So kann untersucht werden, wie es sich auswirkt, ob θ am Rand oder im Zentrum der a-priori-Dichte liegt, und damit auch, wie groß der Einfluss der Wahl der a-priori-Verteilung ist. Datensatz 2 enthält 2 Mutationen und der Watterson-Schätzer ist $\hat{\theta}_W = 0.45$. Die Anzahlen der θ^*, die nicht verworfen werden, sind für die verschiedenen ϵ jeweils in Tabelle 4.2 angeführt.

ϵ	Anzahl
0	256
1	728
2	1202
3	1449
5	1961
10	3159

Tabelle 4.2: Anzahl der akzeptierten θ^* in Datensatz 2

Für die kleinen Werte von ϵ unterscheidet sich die Anzahl der akzeptierten θ^* kaum von der Anzahl für $\theta = 5$, für die größeren Werte sind aber große Unterschiede erkennbar. Bei $\theta = 1$ werden offenbar mehr Werte verworfen.

Die resultierenden a-posteriori-Dichten (Abbildung 4.2) werden mit wachsendem ϵ nicht nur breiter, sondern auch stärker rechts-schief.

4.2 Wahl von ϵ durch 5-fache Kreuzvalidierung

Wie im Kapitel 1.3.6 und im Kapitel 4.1.1 anhand von Beispielen gezeigt wird, beeinflusst die Wahl von ϵ maßgeblich die Form der a-posteriori-Dichte. Eine Methode, die möglicherweise hilfreich sein kann, um den Algorithmus mit dem besten ϵ auszuwählen, ist Kreuzvalidierung. Mit Hilfe einer Simulation soll die Eignung von Kreuzvalidierung hierfür erprobt werden.

Die Simulation basiert wie das Beispiel in Abschnitt 4.1.1 auf dem Standard-Coalescent-Prozess und auch hier wird angenommen, dass die a-posteriori-Dichte von θ geschätzt werden soll.

Als einzige Summary-Statistik für den Rejection-Algorithmus wird $S_h(\boldsymbol{X})$, die Anzahl der Mutationen in einem Datensatz $\boldsymbol{X}$ der Größe h, verwendet. Häufig wird der Einfachheit halber nur S geschrieben.

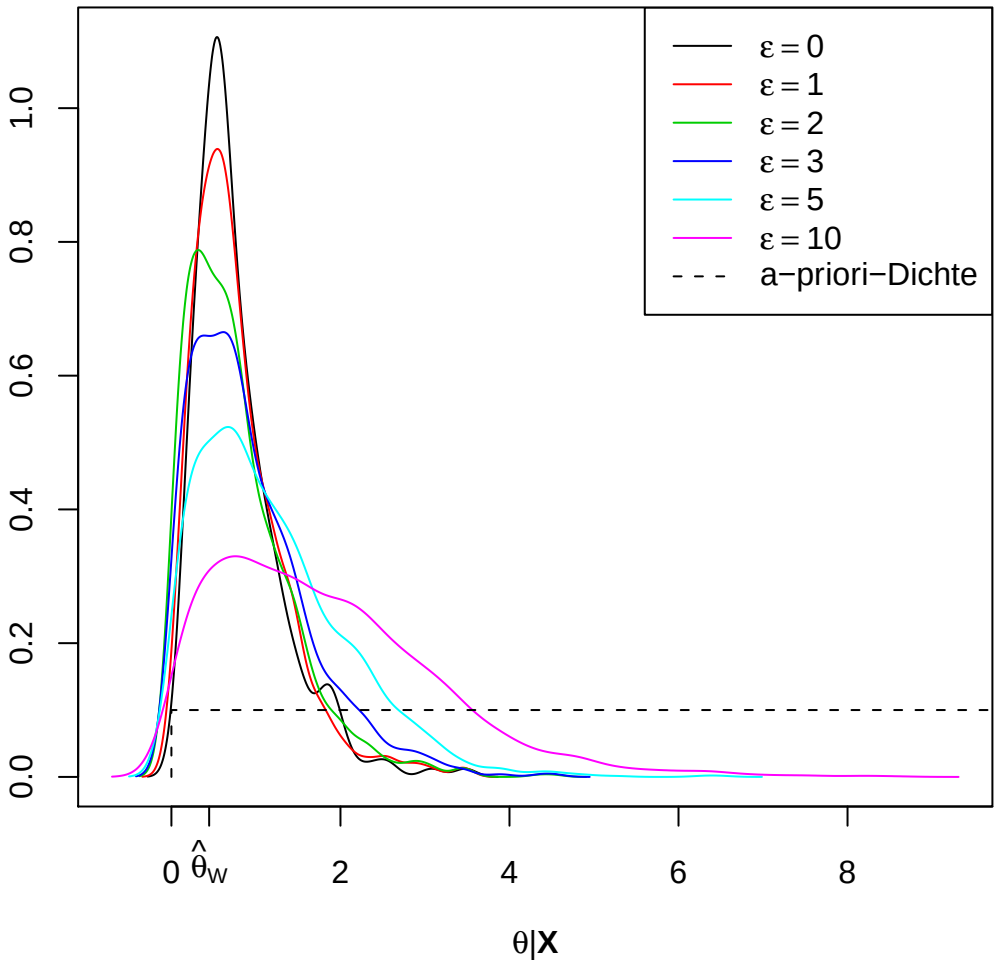


Abbildung 4.2: Mittels ABC simulierte a-posteriori-Dichten von θ in Datensatz 2 bei verschiedenen Werten für ϵ.

Algorithmus 4.2. *θ wird aus $\pi(\theta)$ zufällig gezogen und fixiert. Aus dem Coalescent-Prozess mit Parameter θ wird ein Coalescent-Baum mit $n = 50$ Nachkommen in der letzten Generation simuliert. Die Daten dieser 50 Individuen werden zufällig in 5 gleich große Partitionen geteilt. Eine davon ist das Validierungssample $\boldsymbol{X}_a^{(v)}$, die übrigen vier stellen das Trainingssample $\boldsymbol{X}_a^{(t)}$ dar. Für $a = 1, \ldots, 5$ wird jeweils eine andere Partition zur Validierung verwendet.*

Folgende Schritte werden für $a = 1, \ldots, 5$ durchgeführt:

1. *Aus $\boldsymbol{X}_a^{(t)}$ wird $S_a^{(t)}$ berechnet.*
2. *Rejection-Algorithmus: m-malige Wiederholung der Schritte*
   (a) *Ziehe θ^* aus $\pi(\theta)$.*
   (b) *Simuliere S^* aus dem Coalescent-Prozess mit Parameter θ^*.*
   (c) *Verwirf θ^*, wenn $d\left(S^*, S_a^{(t)}\right) > \epsilon$, sonst speichere θ^* als θ_i^*, wenn θ^* der i-te nicht verworfene simulierte Wert ist.*
3. *Aus $(\theta_i^*)_{1 \le i \le w}$, wobei w die Anzahl der nicht verworfenen Werte ist, wird der Schätzer $\hat{p}_{ABC}(\theta | S_a^{(t)})$ für die a-posteriori-Dichte $p(\theta | \boldsymbol{X}_a^{(t)})$ mit Hilfe eines Kerndichteschätzers berechnet.*
4. *Berechne $S_a^{(v)}$ aus $\boldsymbol{X}_a^{(v)}$.*
5. *Berechne die Kontrastfunktion:*

$$\gamma_a = \gamma\left(\hat{p}_{ABC}\left(\theta \,\middle|\, \boldsymbol{X}_a^{(t)}\right), \boldsymbol{X}_a^{(v)}\right) = -\hat{P}_{ABC}\left(S_a^{(v)} \,\middle|\, S_a^{(t)}\right) \tag{4.1}$$

$$= -\int_\Theta s_{10}\left(S_a^{(v)} \middle|\, \theta\right) \hat{p}_{ABC}\left(\theta \,\middle|\, S_a^{(t)}\right) d\theta, \tag{4.2}$$

wobei $\Theta = [0, \infty)$ (für $s_{10}(\cdot|\theta)$ siehe Gleichung 3.23).

Berechne den Kreuzvalidierungsschätzer für das Risiko der ABC-Dichteschätzung:

$$\hat{\mathcal{L}}^{5F}(\hat{p}_{ABC}, \boldsymbol{X}) = \frac{1}{5} \sum_{a=1}^{5} \gamma_a \tag{4.3}$$

4.2.1 Details

Die Simulation wurde mit R 2.11.1 (R Development Core Team, 2010) und Hudons MS (Hudson, 2002) mit dem Zufallszahlengenerator `rand2` durchgeführt.

Wahl der a-priori-Verteilung

Wie in den Beiträgen von Joyce u. Marjoram (2008), Leuenberger u. a. (2009) und Beaumont u. a. (2002) wird die Gleichverteilung als a-priori-Verteilung für θ gewählt.

Die Gleichverteilung auf einem endlichen Intervall ist zwar keine uninformative a-priori-Verteilung, aber es ist aus verschiedenen Gründen sinnvoll, sie zu verwenden: Erstens kann das Intervall, auf dem sich plausible Werte für θ befinden, relativ leicht bestimmt werden, im Gegensatz zur Verteilung von θ auf diesem Intervall. Zweitens ist es für die a-priori-Verteilung wichtiger, dass sie alle plausiblen Werte für θ enthält, als dass sie um den „wahren“ Wert von θ zentriert ist, da die Daten meist einen größeren Einfluss auf die Form der a-posteriori-Verteilung haben, als die a-priori-Verteilung (Gelman u. a., 2004, S. 39). Drittens sollte, da es sich bei der Simulation um eine experimentelle Anwendung von Kreuzvalidierung im Kontext von ABC handelt, das Modell möglichst einfach und allgemein gehalten werden, damit leichter Vergleiche mit anderen Studien angestellt werden können.

Da $\theta \in \mathbb{R}^+$, wird ein positives Intervall als Träger für die Gleichverteilung gewählt. Joyce u. Marjoram verwenden das Intervall $(0, 10)$, Beaumont u. a. $(0, 50)$ und Leuenberger u. a. $(0.05, 10)$. Hier wird das Intervall bei $(0, 10)$ festgesetzt. Die Simulation wird für $\theta = 5$ durchgeführt.

Dichteschätzung

Zur Dichteschätzung wird ein Kerndichteschätzer mit Normalverteilungskern verwendet. Die Bandbreite wird mit der default-Einstellung der Funktion `density` in R berechnet (`bw = "nrd0"`), die Silvermans „Rule of Thumb“ verwendet. Eine genauere Beschreibung kann in R Development Core Team (2010) gefunden werden, insbesondere in der Beschreibung der Funktion `bw.nrd`, mit der `density` die Bandbreite berechnet.

Distanzmaß

Da es sich bei S um eine Anzahl handelt, wird als Distanzmaß wird der Betrag verwendet. $d(S, S^*)$ ist daher immer eine nicht-negative ganze Zahl.

Simulation von $\boldsymbol{X}$ und S

Das C-Programm MS von Richard R. Hudson (beschrieben in Hudson (2002)) simuliert Coalescent-Bäume unter dem Infinite-Sites-Modell. Dabei ist es möglich, Rekombination, Migration und verschiedene demographische Szenarien zu modellieren. Um ein Sample aus dem Standard-Coalescent-Prozess

zu generieren, müssen nur die skalierte Mutationsrate θ und die Größe der Stichprobe n eingegeben werden.

MS wird über eine Command-Shell bedient. Im einfachsten Fall sieht der Befehl folgendermaßen aus:

```
ms 10 2 -t 5.0 > outfile
```

So werden zwei Stichproben der Größe 10 mit $\theta = 5.0$ erzeugt. Das Ergebnis, das in die Datei `outfile` geschrieben wird:

```
ms 5 2 -t 5.0
28939

//
segsites: 24
positions: 0.0820 0.1276 0.1810 0.2162 0.2464 0.3285 0.3322
0.3437 0.3660 0.3762 0.3835 0.3988 0.4185 0.4349 0.5454
0.5901 0.6189 0.7052 0.7710 0.7776 0.7916 0.8527 0.9894
0.9903
000010111100000100000100
001100000001101011110011
000010111110000100000100
001100000001101011110011
111101000000110010101000

//
segsites: 11
positions: 0.0244 0.1203 0.2466 0.2534 0.2869 0.3716 0.6573
0.7073 0.7212 0.7983 0.8142
01101001000
00010110101
10010110101
00010110011
00010110101
```

Nach der Wiederholung des eingegeben Befehls folgt der Seed, mit dem die Zufallszahlen für den Prozess erzeugt wurden.

`segsites` gibt an, an wievielen Positionen auf der DNA-Sequenz Mutationen stattgefunden haben, `positions`, wo sie stattgefunden haben, wobei die Länge der DNA-Sequenz auf das Intervall $(0, 1)$ skaliert ist. Anschließend steht für jedes Individuum der Stichprobe eine Zeile: An den Stellen mit Eintrag 0 hat seit dem MRCA keine Mutation stattgefunden, an den Stellen mit 1 schon.

Es kann auch ein beliebiger Seed eingegeben werden, hier zum Beispiel 1, um wiederholt denselben Datensatz erzeugen zu können:

```
ms 10 2 -t 5.0 -seeds 1 > outfile
```

Mit der Funktion `system` können in R unter Windows Befehle an die Windows-Command-Shell gesandt werden, so kann MS von R aus verwendet werden.

Im Algorithmus 4.2 wird nur der Grunddatensatz mittels MS aus einem Coalescent-Baum simuliert, im Schritt 2(b) wird die Anzahl der Mutationen S_{40} direkt aus ihrer Verteilung $\mathcal{S}_{40}$ simuliert (siehe Abschnitt 3.2.5). Der Grund dafür ist, dass aus dem Grunddatensatz zufällig Individuen für das Validierungssample gezogen werden müssen. Es wäre nicht zulässig, diesen Schritt zu übergehen und stattdessen für das Trainings- und das Validierungssample aus $\mathcal{S}_{40}$ bzw. aus $\mathcal{S}_{10}$ zu ziehen, da das Trainings- und das Validierungssample nicht unabhängig von einander sind. Da im Schritt 2(b) diese Komplikation entfällt, kann direkt aus $\mathcal{S}_{40}$ gezogen werden. Das erfolgt, indem zuerst die Gesamtlänge der Äste des Baumes als Summe über exponentialverteilte Zufallsvariablen simuliert wird (siehe Abschnitt 3.2.5) und anschließend die Poisson-verteilte Anzahl der Mutationen für einen Baum dieser Länge (siehe Abschnitt 3.2.5).

Wahl der Kontrastfunktion und Berechnung der Prognose-Verteilung

Als Kontrastfunktion wird

$$\gamma\left(\hat{p}_{ABC}\left(\theta\,\middle|\,\boldsymbol{X}^{(t)}\right),\boldsymbol{X}^{(v)}\right)=-P\left(S^{(v)}\,\middle|\,S^{(t)}\right), \tag{4.4}$$

verwendet.

γ ist der Prognose-Dichte $p(\boldsymbol{X}^{(v)}|\boldsymbol{X}^{(t)})$ ähnlich (siehe Abschnitt 1.1). Der Einfachheit halber werden aber die Daten $\boldsymbol{X}^{(v)}$ bzw. $\boldsymbol{X}^{(t)}$ durch die Anzahl der Simulationen $S^{(v)}$ und $S^{(t)}$ ersetzt, da die Berechnung der Prognose-Dichte die Berechnung der Likelihood $P(\theta|\boldsymbol{X}^{(t)})$ voraussetzt.

Nach 1.2 gilt:

$$P\left(S^{(v)}\,\middle|\,S^{(t)}\right)=\int_{\Theta}p(S^{(v)}|\theta)p(\theta|S^{(t)})d\theta \tag{4.5}$$

$p(\theta|S^{(t)})$ wird durch den beschriebenen Rejection-Algorithmus und eine Kerndichteschätzung geschätzt, $p(S^{(v)}|\theta)$ ist bekannt (Abschnitt 3.2.5). Die Implementierung in R sieht folgendermaßen aus:

```
S.predictive = function(Sv, St, nv, nt, theta.post, anz = 512)
{
dens.post = density(theta.post, n = anz, from = 0.05)  # 1)
x = dens.post$x   # Stützstellen, an denen die Dichte geschätzt
# wird
x = as.matrix(x)
y = dens.post$y   # Dichte an den Stützstellen
likelihood = apply(x, 1, S.density, nv, Sv)   # 2)
mean(likelihood*y*(x[anz]-x[1]))  # 3)
}
```

Die Parameter der Funktion `S.predictive` sind die Anzahl der Mutationen im Validierungs- und im Trainingssample (`Sv` und `St`), die Größe der Samples (`nv` und `nt`), die simulierten Werte θ^*, die akzeptiert wurden (`theta.post`), sowie die Anzahl der Stützstellen (`anz`), an denen die a-posteriori-Dichte geschätzt werden soll.

ad 1): Die a-posteriori-Dichte $p(\theta|S^{(t)})$ wird erst ab $\theta = 0.05$ geschätzt, da $\theta \leq 0$ zu falschen Werten in der Berechnung der Likelihood (weiter unten) führt. `density` berechnet den Kerndichteschätzer an `n = anz` Stützstellen. Wenn `n > 512`, wird für die Berechnung auf die nächstgrößere Zweierpotenz aufgerundet und die Dichte wird anschließend für die gewünschten Stützstellen interpoliert. Es ist also sinnvoll, eine Zweierpotenz zu wählen (R Development Core Team, 2010, Beschreibung der Funktion `density`).

ad 2): $p(S^{(v)}|\theta)$ wird für alle Stützstellen von θ berechnet. Die verwendete Funktion `S.density` wird weiter unten beschrieben.

ad 3): Da `density` die Dichte an `anz` Stützstellen numerisch auswertet, kann das Integral in Gleichung 4.5 nur näherungsweise berechnet werden. Es seien $\theta_1 < \ldots < \theta_{anz}$ die Werte von θ, an denen `density` die a-posteriori-Dichte von θ berechnet. Das Integral wird durch eine Summe angenähert:

$$\int_\Theta s(S^{(v)}|\theta)\hat{p}_{ABC}(\theta|S^{(t)})d\theta \approx \frac{\theta_{anz} - \theta_1}{anz} \sum_{i=1}^{anz} s(S^{(v)}|\theta_i)\hat{p}_{ABC}(\theta_i|S^{(t)}), \quad (4.6)$$

wobei $s()$ die Wahrscheinlichkeitsfunktion von S ist (Abschnitt 3.2.5). Die Genauigkeit der Approximation kann durch die Wahl von `anz` bestimmt werden.

Die verwendete Funktion `S.density` berechnet $p(S^{(v)}|\theta)$ nach der Wahrscheinlichkeitsfunktion in Gleichung 3.23:

```
S.density = function(theta,n,k){
i = 1:(n-1)
hilfsvec = (-1)^(i-1)*choose(n-2, i-1)*(theta/(i+theta))^(k+1)
```

```
hilfssum.pos = sum (hilfsvec[hilfsvec>=0])
hilfssum.neg = sum (hilfsvec[hilfsvec<0])
hilfssum = hilfssum.pos + hilfssum.neg
((n-1)/theta)*hilfssum
}
```

Die Parameter der Funktion sind θ, die Stichprobengröße `n` und die Anzahl der Mutationen `k`. Da die Werte in `hilfsvec` sehr unterschiedliche Größenordnungen haben, werden, um numerische Fehler zu vermeiden, die positiven und negativen Werte aus `hilfsvec` zuerst getrennt addiert.

Kreuzvalidierung

Damit der rechnerische Aufwand nicht zu groß ist und das Experiment an mehreren Datensätzen durchgeführt werden kann, wurde 5-fache Kreuzvalidierung bzw. 100-malige Wiederholung davon gewählt.

Es kann nicht angenommen werden, dass es sich beim ABC-Schätzer für die a-posteriori-Dichte um einen Minimum-Kontrast-Schätzer über den Kontrast γ handelt, da bayesianische Inferenz nicht auf dem Minimum-Kontrast-Prinzip beruht. Weiters sind Trainings- und Validierungssample nicht unabhängig. Das heißt, dass die in Kapitel 2.4 beschriebenen Eigenschaften des Kreuzvalidierungsschätzers nicht gelten. Es ist nicht klar, ob ein Kontrast existiert, für den der ABC-Schätzer ein Minimum-Kontrast-Schätzer ist.

Summary-Statistik

Als Summary-Statistik wird S, die Anzahl der Mutationen gewählt. Einerseits aus praktischen Gründen: Die Verteilung von S ist bekannt und es kann daher ohne Simulation von DNA-Daten simuliert werden. Andererseits ist S eine lineare Transformation des Watterson-Schätzers $\hat{\theta}_W$, der zwar nicht suffizient ist, aber laut Joyce u. Marjoram (2008) „nearly sufficient“.

Kapitel 5

Ergebnisse

5.1 5-fache Kreuzvalidierung

Es werden 50 Datensätze der Größe $n = 50$ mit den Seeds 1 bis 50 und $\theta = 5$ simuliert. Die Anzahl der Wiederholungen des Rejection-Algorithmus beträgt $m = 10000$. Das Risiko des Schätzers $\hat{p}_{ABC}(\theta|S)$ für die a-posteriori-Dichte $p(\theta|\boldsymbol{X})$ wird mit 5-facher Kreuzvalidierung, wie in Algorithmus 4.2 beschrieben, geschätzt. Das heißt, die Größe des Trainingssamples beträgt $n_t = 40$ und die Größe des Validierungssamples $n_v = 10$. Die θ^* werden für jeden Durchgang der Kreuzvalidierung einmal simuliert, anschließend werden aus diesem Vektor Subsets für die einzelnen Werte von ϵ gebildet.

Da $d(S^*, S)$ immer eine nicht-negative ganze Zahl ist, wird $\epsilon = 0, 1, \ldots, 15$ gewählt.

Abbildung 5.1 lässt vermuten, dass das geschätzte Risiko leicht positiv von ϵ abhängt, doch dieser Trend ist im Vergleich zur Streuung des Risikos eher schwach und nur bei den niedrigen Risiko-Werten erkennbar.

In Abbildung 5.2 ist zu sehen, dass das Risiko stärker vom Datensatz als von ϵ abhängt.

Damit das Risiko dennoch als Auswahlkriterium für ϵ verwendet werden kann, muss entweder ein Mittelwert über viele Datensätze gebildet werden, oder es wird nur der Datensatz, über den Inferenz betrieben werden soll, untersucht. Dann ist es sinnvoll, nicht nur v-fache Kreuzvalidierung, sondern wiederholte Kreuzvalidierung oder Leave-p-out-Kreuzvalidierung zu verwenden, um einen möglichst genauen Schätzer zu erhalten, der nicht von der zufälligen Partitionierung in der 5-fachen Kreuzvalidierung abhängt.

Betrachtet man das Risiko gemittelt über die 50 Datensätze (Abbildung 5.3), liegt das minimale Risiko bei $\epsilon = 1$. Wenn also das minimale durchschnittliche Risiko als Kriterium gewählt wird, um ϵ festzulegen, wird $\epsilon = 1$

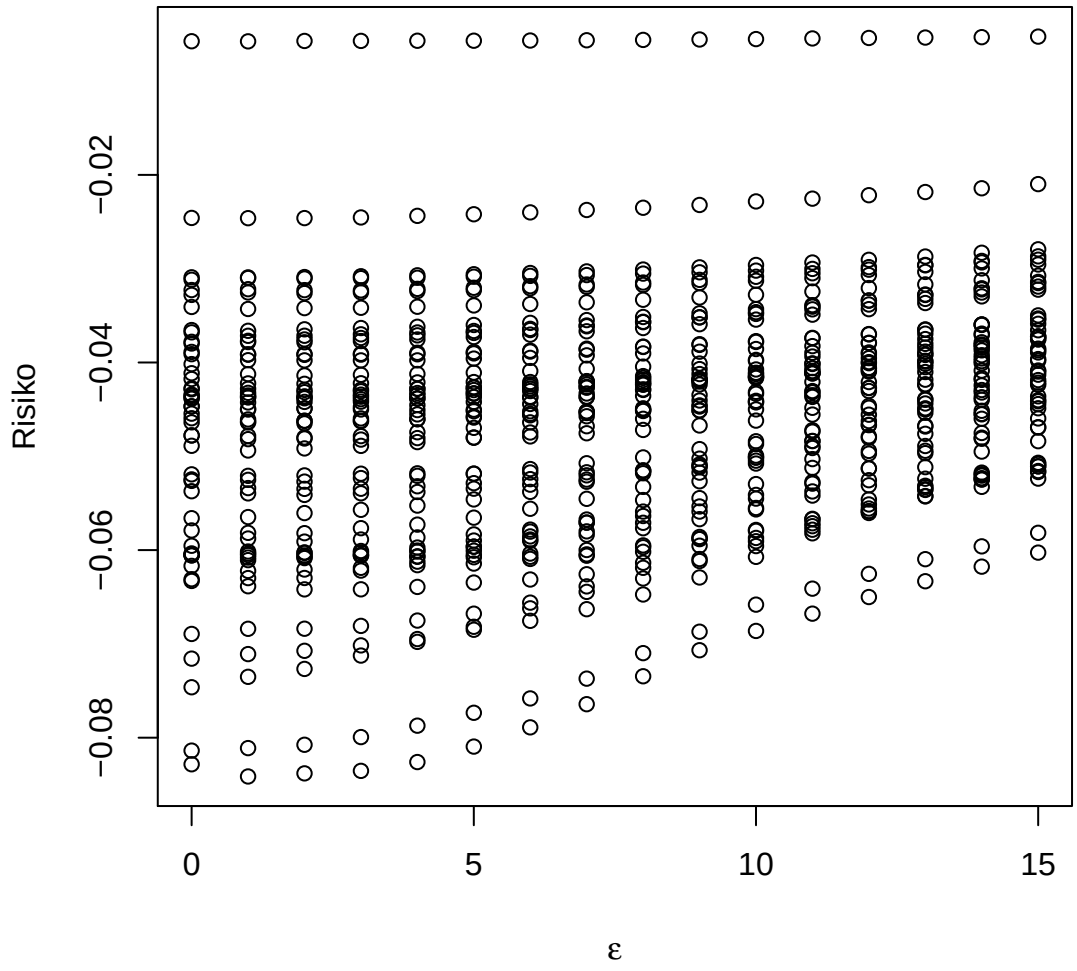


Abbildung 5.1: Mit 5-facher Kreuzvalidierung geschätztes Risiko von $\hat{p}_{ABC}$ für 50 Datensätze in Abhängigkeit von ϵ

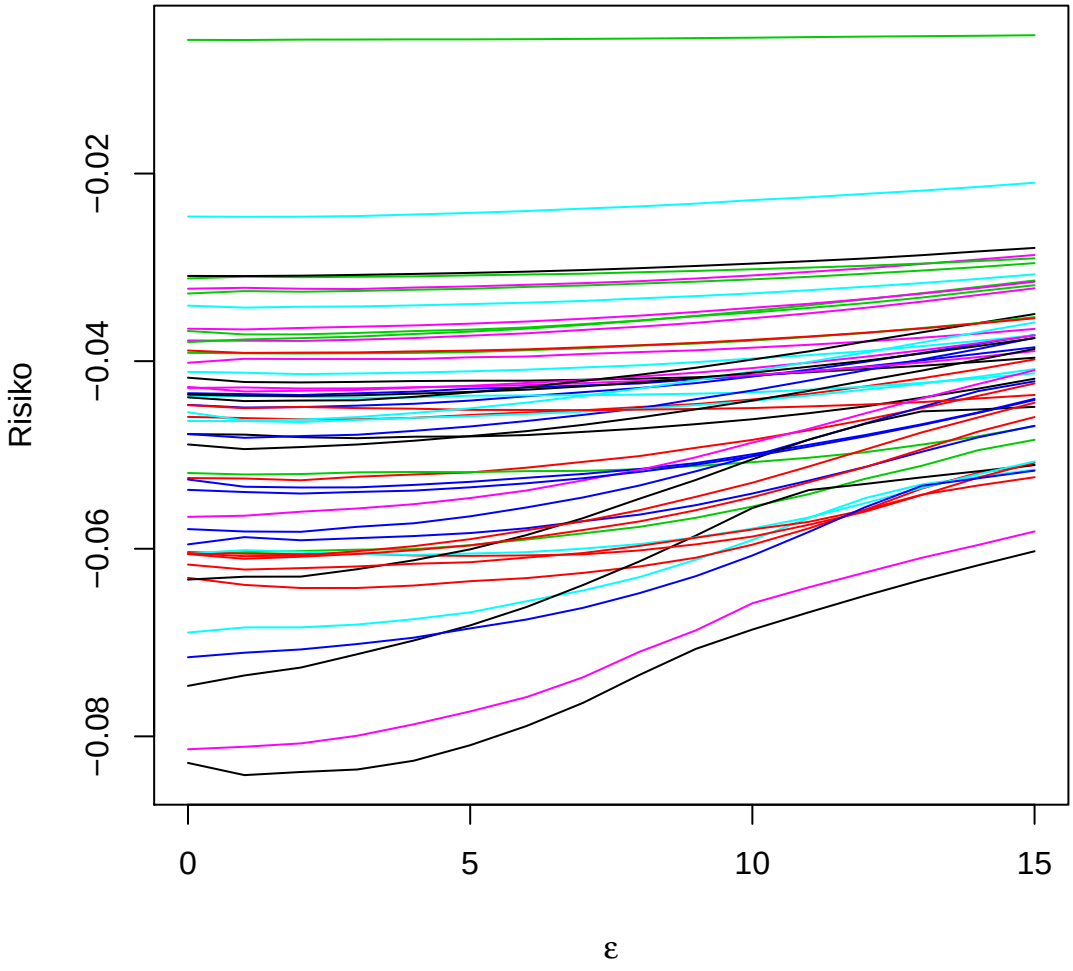


Abbildung 5.2: Mit 5-facher Kreuzvalidierung geschätztes Risiko von $\hat{p}_{ABC}$ in Abhängigkeit von ϵ. Jede Linie stellt das Risiko eines Datensatzes dar. Manche Linien beginnen erst bei $\epsilon = 0.2$, da für $\epsilon = 0.1$ bei mindestens einem Durchgang der Kreuzvalidierung nicht genügend θ^* akzeptiert wurden, um das Risiko zu schätzen.

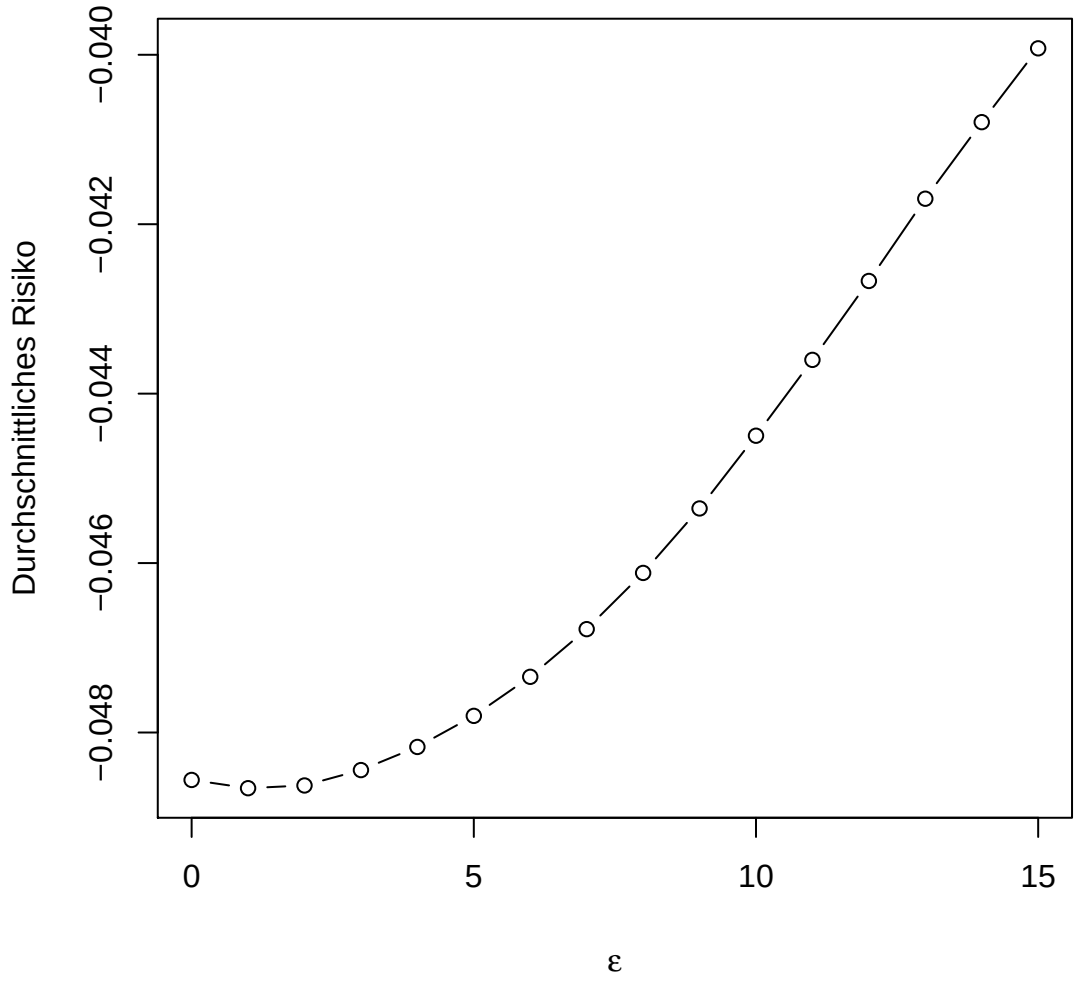


Abbildung 5.3: Mit 5-facher Kreuzvalidierung geschätztes Risiko von $\hat{p}_{ABC}$ für 50 Datensätze in Abhängigkeit von ϵ, Durchschnitt über 50 Datensätze in Abhängigkeit von ϵ

ausgewählt.

In Abbildung 5.4 ist dargestellt, wie häufig die verschiedenen Werte von ϵ in den einzelnen Datensätzen das Risiko minimieren. Diese Verteilung gibt an, welche ϵ's ausgewählt werden, wenn das minimale Risiko des einzelnen Datensatzes, der untersucht wird, zur Wahl von ϵ herangezogen wird.

ϵ	Minimum	Mittelwert	Maximum
0	35.8	233.2	268.2
1	93.4	693.1	782.6
2	154.6	1151.4	1298.0
3	214.2	1607.7	1795.4
4	278.6	2061.2	2305.8
5	346.8	2514.4	2820.4

Tabelle 5.1: Anzahl der akzeptierten θ^* für $\epsilon = 0, 1, \ldots, 5$. Die Grundlage bildet die durchschnittliche Anzahl der akzeptierten θ^* über die 5 Wiederholungen der Kreuzvalidierung.

In Tabelle 5.1 sind die durchschnittlichen Anzahlen der nicht verworfenen θ^* für $\epsilon \leq 5$ angeführt. Schon bei $\epsilon = 0$ ist der Mittelwert 233.22, damit ist eine Kerndichteschätzung sinnvoll. Das Minimum liegt allerdings weit darunter (35.8). Bedenkt man, dass alle Werte in Tabelle 5.1 aus den durchschnittlichen Anzahlen über die 5 Wiederholungen der Kreuzvalidierung berechnet werden, ist anzunehmen, dass es auch noch niedrigere Anzahlen gibt, bei denen eine Kerndichteschätzung keinesfalls sinnvoll ist.

Ab $\epsilon = 1$ ist aber anzunehmen, dass eine Kerndichteschätzung in den meisten Fällen unproblematisch ist.

5.2 Wiederholte 5-fache Kreuzvalidierung

Mit einer erneuten Simulation, bei der wieder dieselben Datensätze verwendet werden, wird untersucht, ob sich die Wahl von ϵ mittels 100-mal wiederholter 5-facher Kreuzvalidierung auf weniger unterschiedliche Werte von ϵ konzentriert, als bei einfacher 5-facher Kreuzvalidierung. Der übrige Algorithmus wird nicht verändert.

Das Gesamtbild ändert sich kaum (5.5), auch nicht die durchschnittlichen Risiko-Werte (5.6).

Mit Abstand am häufigsten wird $\epsilon = 1$ ausgewählt (Abbildung 5.7).

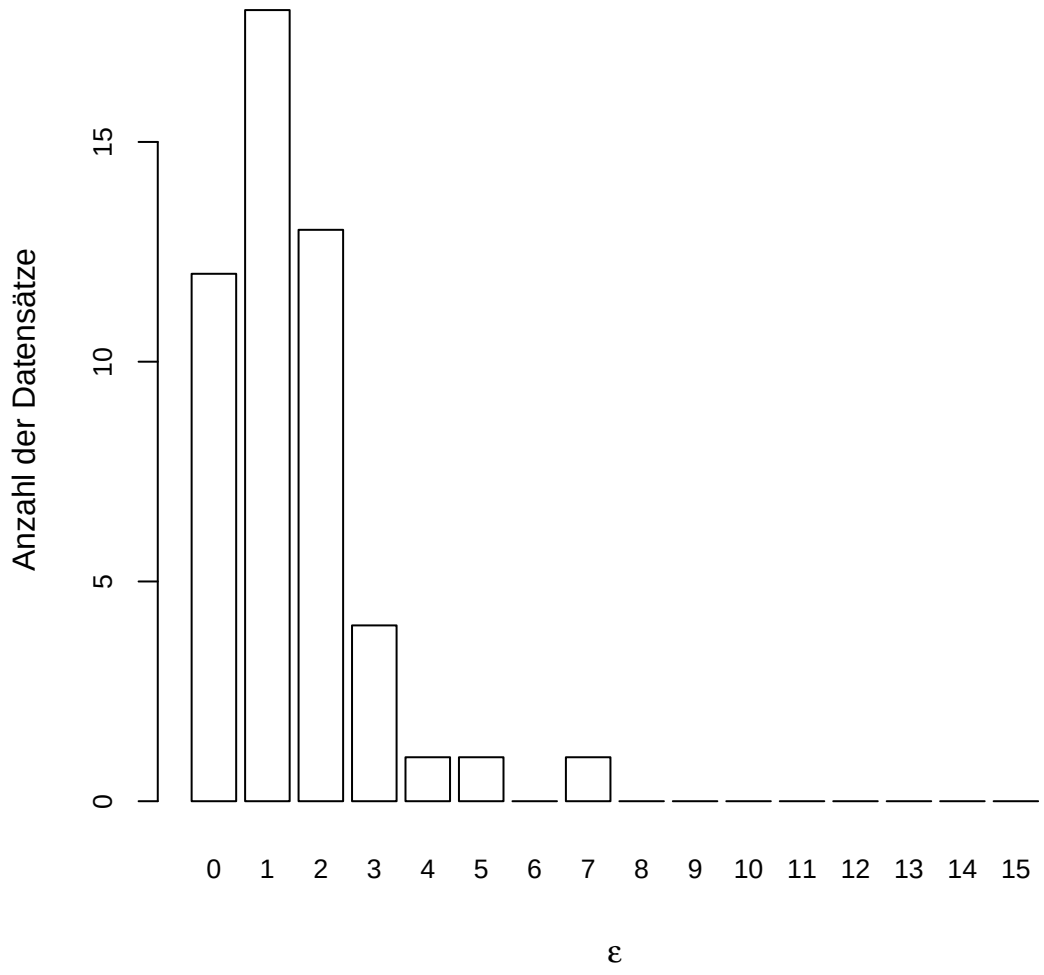


Abbildung 5.4: Anzahl der Datensätze, für die das jeweilige ϵ das geschätzte Risiko minimiert

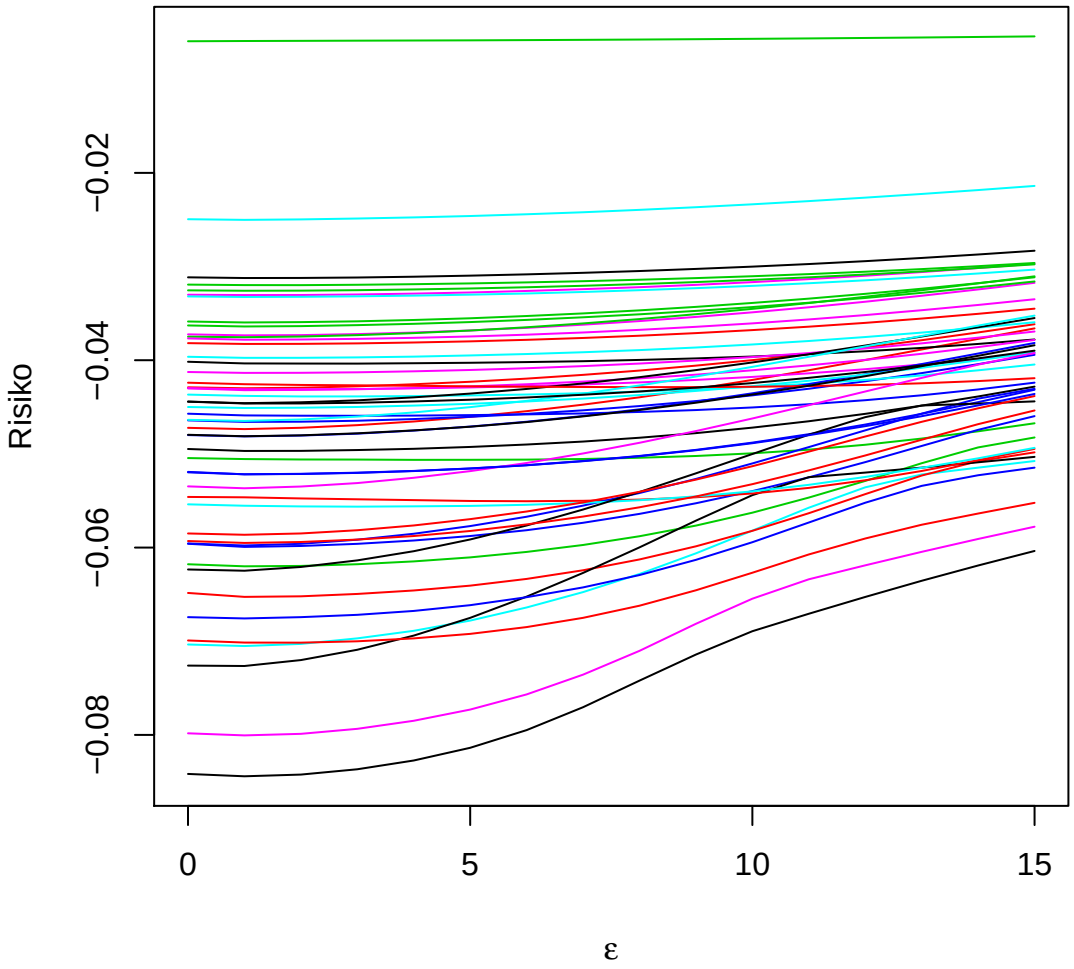


Abbildung 5.5: Mit 100-mal wiederholter 5-facher Kreuzvalidierung geschätztes Risiko von $\hat{p}_{ABC}$ für 50 Datensätze in Abhängigkeit von ϵ. Jede Linie stellt das Risiko eines Datensatzes dar.

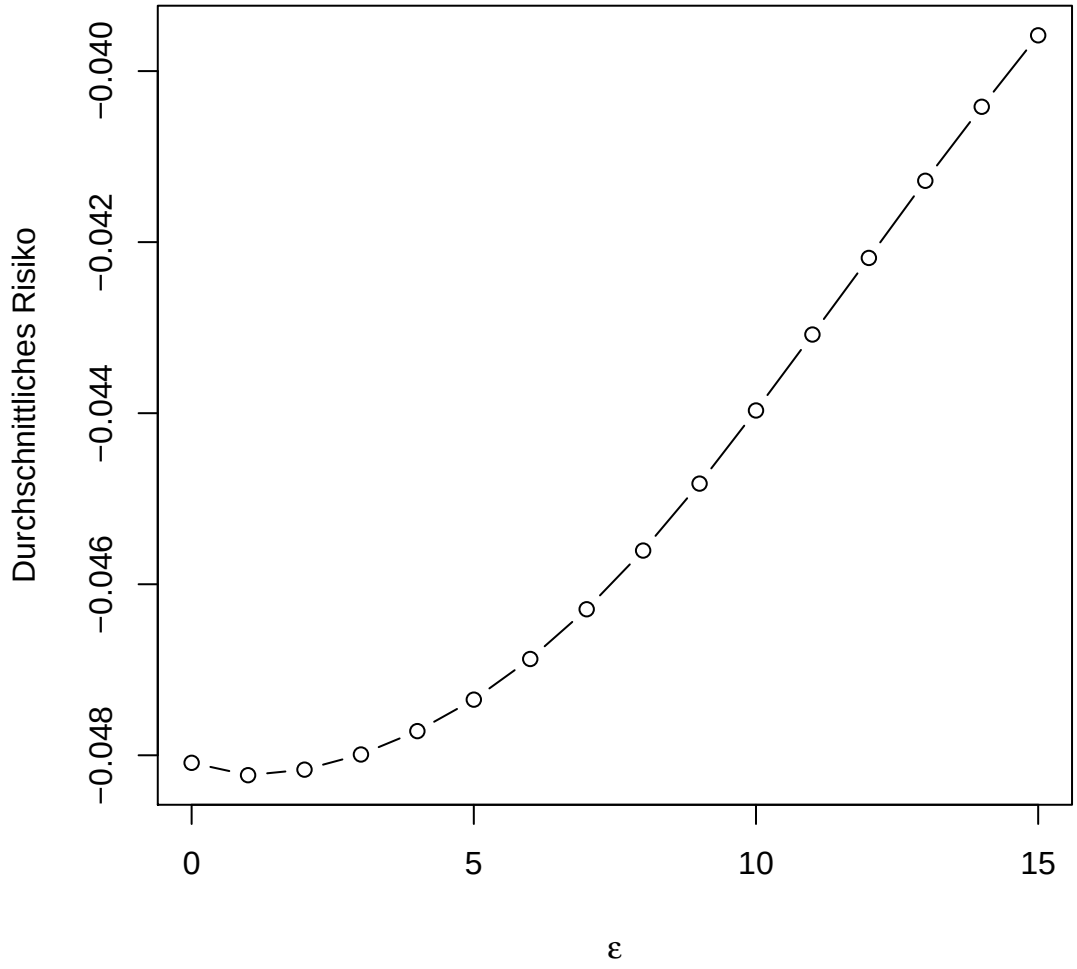


Abbildung 5.6: Mit 100-mal wiederholter 5-facher Kreuzvalidierung geschätztes Risiko von $\hat{p}_{ABC}$ für 50 Datensätze in Abhängigkeit von ϵ, Durchschnitt über 50 Datensätze in Abhängigkeit von ϵ

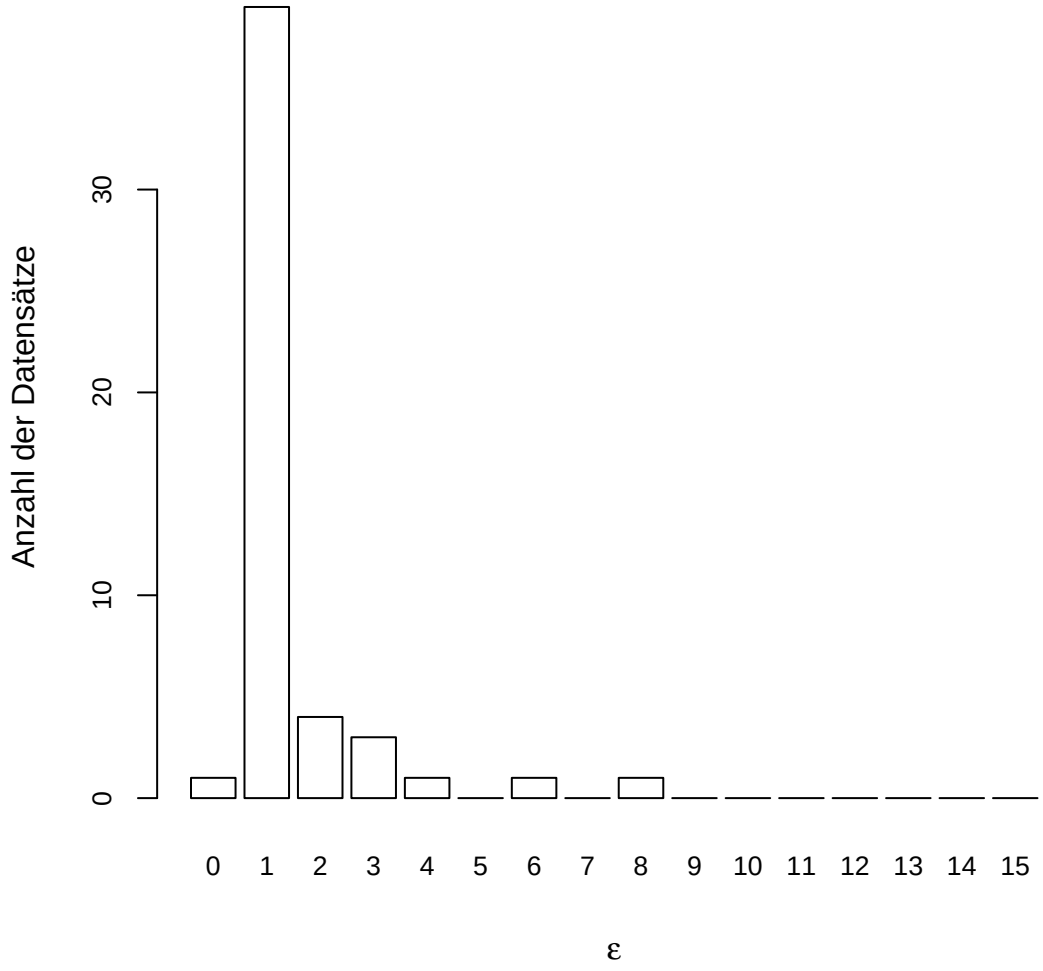


Abbildung 5.7: Anzahl der Datensätze, für die das jeweilige ϵ das mit 100-mal wiederholter 5-facher Kreuzvalidierung geschätzte Risiko minimiert

Zusammenfassung und Ausblick

Der erste Teil der Diplomarbeit beschäftigt sich auf theoretischer Ebene mit drei Themen: Im ersten Kapitel wird Approximate Bayesian Computation (ABC) definiert und theoretisch behandelt, wobei besonders auf die Wahl von ϵ, die Wahl der Summary-Statistiken und die von Beaumont u. a. (2002) vorgeschlagene Regressionskorrektur eingegangen wird. Das zweite Kapitel beschäftigt sich mit Kreuzvalidierung, hauptsächlich aus einer klassischen statistischen Perspektive, und beschreibt, wie Modell- bzw. Algorithmenwahl mittels Kreuzvalidierung funktioniert. Im letzten Theorie-Kapitel, dem dritten Kapitel, wird der Coalescent-Prozess aus dem Wright-Fisher-Modell hergeleitet und seine Eigenschaften und Erweiterungen werden kurz dargelegt.

Im zweiten Teil der Arbeit werden diese drei Themen, die bis dahin nur getrennt von einander behandelt wurden, in einer Simulationsstudie zusammengeführt: Es soll anhand von DNA-Daten, die aus einem Coalescent-Modell simuliert werden, ausprobiert werden, ob Kreuzvalidierung dafür verwendet werden kann, ein geeignetes ϵ für einen ABC-Algorithmus zu finden. Im Kapitel vier wird die Simulationsstudie ausführlich beschrieben, die Ergebnisse werden in Kapitel fünf präsentiert.

Besonders die Abbildung 5.7 zeigt, dass Kreuzvalidierung grundsätzlich geeignet sein kann, um ϵ auszuwählen, möglicherweise auch Summary-Statistiken. Um das genauer zu untersuchen, wäre es aber wichtig, auch Beispiele mit bekannter a-posteriori-Verteilung zu verwenden, sodass das Ergebnis, das mit dem ϵ aus der Kreuzvalidierung erzielt wird, mit der wahren Verteilung verglichen werden kann.

Worauf in Zukunft noch genauer eingegangen werden könnte, ist die Kontrastfunktion. Möglicherweise ist es von Vorteil, die Anzahl der akzeptierten θ^*'s miteinzubeziehen, sodass auf jeden Fall eine gewisse Mindestanzahl erreicht werden muss, damit die Kerndichteschätzung Sinn macht. Das könnte alternativ auch über einen Parameter der Kerndichteschätzung geschehen. Auch andere Kreuzvalidierungsalgorithmen sollten noch getestet werden, wobei aber Leave-p-out in vielen Fällen nicht praktikabel ist. Einfache 5-fache Kreuzvalidierung ist aber zu stark vom Zufall abhängig, als dass es geeignet

wäre, um einen einzigen Datensatz zu behandeln. Die Ergebnisse der 100-mal wiederholten 5-fachen Kreuzvalidierung sind in diesem ersten Experiment zufriedenstellend, aber es wäre trotzdem sinnvoll, noch andere Methoden auszuprobieren und auch theoretische Überlegungen dazu anzustellen, welcher Kreuzvalidierungsschätzer in diesem Bayesianischen Kontext mit abhängigen Datensätzen die besten Ergebnisse erzielen kann.

Insgesamt sind die Ergebnisse vielversprechend: Kreuzvalidierung könnte eine geeignete Methode sein, um die Parameter eines ABC-Algorithmus zu bestimmen, es wird aber notwendig sein, noch weitere Simulationen und theoretische Überlegungen anzustellen.

Literaturverzeichnis

[Alqallaf u. Gustafson 2001] Alqallaf, Fatemah ; Gustafson, Paul: On Cross-Validation of Bayesian Models. In: *The Canadian Journal of Statistics* 29 (2001), S. 333–340

[Arlot u. Celisse 2010] Arlot, Sylvain ; Celisse, Alain: A survey for cross-validation procedures for model selection. In: *Statistics Surveys* 4 (2010), S. 40–79

[Barankin u. Maitra 1963] Barankin, Edward W. ; Maitra, Ashok P.: Generalization of the Fisher-Darmois-Koopman-Pitman Theorem on Sufficient Statistics. In: *Sankhyā: The Indian Journal of Statistics, Series A* 25 (1963), S. 217–244

[Beaumont u. a. 2002] Beaumont, Mark A. ; Zhang, Wenyang ; Balding, David J.: Approximate Bayesian Computation in population genetics. In: *Genetics* 162 (2002), S. 2025–2035

[Bickel u. Doksum 2006] Bickel, Peter J. ; Doksum, Kjell A.: *Mathematical Statistics*. Zweite Ausgabe. Prentice Hall, 2006

[Blum 2010] Blum, Michael G. B.: Choosing the Summary Statistics and the Acceptance Rate in Approximate Bayesian Computation. In: Saporta, G. (Hrsg.) ; Lechevaille, Y. (Hrsg.): *COMPSTAT 2010 – Proceedings in Computational Statistics*, 2010

[Blum u. François 2010] Blum, Michael G. B. ; François, Olivier: Non-linear regression models for Approximate Bayesian Computation. In: *Statistics and Computing* 20 (2010), S. 63–73

[Carlin u. Louis 1996] Carlin, Bradley P. ; Louis, Thomas A.: *Bayes and Empirical Bayes Methods for Data Analysis*. Erste Ausgabe. Chapman & Hall, 1996

[Felsenstein 2006] Felsenstein, Joseph: Accuracy of Coalescent Likelihood Estimates: Do We Need More Sites, More Sequences or More Loci? In: *Molecular Biology and Evolution* 23 (2006), S. 691–700

[Gelman u. a. 2004] Gelman, Andrew ; Carlin, John B. ; Stern, Hal S. ; Rubin, Donald B.: *Bayesian Data Analysis*. Zweite Ausgabe. Chapman & Hall, 2004

[Hartl u. Clark 2007] Hartl, Daniel L. ; Clark, Andrew G.: *Principles of Population Genetics*. Vierte Ausgabe. Palgrave Macmillan, 2007

[Hein u. a. 2005] Hein, Jotun ; Schierup, Mikkel H. ; Wiuf, Carsten: *Gene Genealogies, Variation and Evolution*. Oxford University Press, 2005

[Hudson 2002] Hudson, Richard R.: Generating Samples Under a Wright-Fisher Neutral Model of Genetic Variation. In: *Bioinformatics* 18 (2002), S. 337–338

[Joyce u. Marjoram 2008] Joyce, Paul ; Marjoram, Paul: Approximately sufficient statistics and bayesian computation. In: *Statistical Applications in Genetics and Molecular Biology* 7 (2008), Nr. 1

[Kingman 1982] Kingman, J. F. C.: On the Genealogy of Large Populations. In: *Journal of Applied Probability* 19 (1982), S. 27–43

[Leuenberger u. a. 2009] Leuenberger, Christoph ; Wegmann, Daniel ; Excoffier, Laurent: *Bayesian computation and model selection in population genetics*. 2009. – arXiv:0901.2231v1

[Marjoram u. Tavaré 2006] Marjoram, Paul ; Tavaré, Simon: Modern computational approaches for analysing molecular genetic variation data. In: *Nature Reviews Genetics* 7 (2006), S. 759–770

[Nordborg 2007] Nordborg, Magnus: Coalescent Theory. In: Balding, David J. (Hrsg.) ; Bishop, Martin (Hrsg.) ; Cannings, Chris (Hrsg.): *Handbook of Statistical Genetics* Bd. 2. Dritte Ausgabe. John Wiley & Sons, 2007, S. 843–877

[R Development Core Team 2010] R Development Core Team: *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2010. `http://www.R-project.org`

[Robert 2007] Robert, Christian P.: *The Bayesian Choice*. Zweite Ausgabe. Springer, 2007

[Stephens 2007] Stephens, Matthew: Inference Under the Coalescent. In: Balding, David J. (Hrsg.) ; Bishop, Martin (Hrsg.) ; Cannings, Chris (Hrsg.): *Handbook of Statistical Genetics* Bd. 2. Dritte Ausgabe. John Wiley & Sons, 2007, S. 878–908

[Tavaré u. a. 1997] Tavaré, Simon ; Balding, David J. ; Griffiths, R. C. ; Donnelly, Peter: Inferring coalescence times from DNA sequence data. In: *Genetics* 145 (1997), S. 505–518

[Wakeley 2008] Wakeley, John: *Coalescent Theory*. Erste Ausgabe. Roberts & Company Publishers, 2008

Lebenslauf

Persönliche Daten

Johanna Bertl, Bakk. rer. soc. oec.

geboren am 2. Dezember 1985 in Wörgl, Tirol

Ausbildung

09/92–07/96	Volksschule Augasse, Bregenz
09/96–07/04	Bundesgymnasium Gallusstraße, Bregenz
10/04–02/08	Bakkalaureatsstudium aus Statistik, Universität Wien
03/08–08/10	Magisterstudium aus Statistik, Universität Wien

Arbeitserfahrung

10/07–06/09	Tutorin für Statistik an der Fakultät für Wirtschaftswissenschaften und der Fakultät für Informatik der Universität Wien
09/09–03/10	Projektassistentin am Institut für Ökonometrie der Technischen Universität Wien

English Abstract

Approximate Bayesian Computation

Approximate Bayesian Computation (ABC) is a simulation method to find the posterior distribution of a parameter θ of a model $\mathcal{M}$, in cases where the likelihood $p(x|\theta)$ cannot be calculated analytically.

A standard ABC algorithm consists of the following steps, which are repeated m times:

1. Simulate θ^* from the prior distibution $\pi(\theta)$.
2. Simulate data X from $\mathcal{M}(\theta^*)$.
3. Compute the summary statistic S^* from the simulated data and compare it to S, the summary statistic from the real data. If $d(S^*, S) \leq \epsilon$, keep θ^*, else, reject it.

From the θ^*, which were accepted, the posterior density can be estimated, e.g. with a kernel density estimator.

This algorithm raises two questions:

- Which summary statistic(s) S are appropriate? If no sufficient statistics exist, it is not trivial to identify the statistics that carry the most information about a parameter. High dimensional summary statistics may be very informative, but at the same time they make the proportion of accepted θ^*'s very small.
- How shall ϵ be chosen? The smaller ϵ is, the better the estimated posterior density approximates the true posterior density. In practice, however, a small ϵ means that only very few θ^* are accepted, so the stochastic noise in the estimation of the density is very high.

The Coalescent

The coalescent is a widely used stochastic process to analyse genetic variation in population genetics. It is an important area of application of ABC, because the likelihood function of the parameters, that determine the shape of the genealogical tree of n individuals in the last generation, can only be worked out with a tremendous computational effort. Even for the usually small sample sizes of $n \leq 50$ the likelihood cannot be computed in a reasonable time.

A further aspect that makes the coalescent an interesting field for ABC is that for the parameters of a coalescent model, there usually are no sufficient statistics.

The Role of Cross-Validation

By now, as ABC is quite a "young"method, not many approaches exist to answer the open questions. As cross-validation is a method to estimate the risk of a statistical algorithm, it may be a useful tool to choose among ABC algorithms with different values of ϵ and different summary statistics.

Results

The application of 5-fold cross-validation and 100 times repeated 5-fold cross-validation on data which was simulated from a coalescent model showed that cross-validation might be an appropriate tool to choose ϵ. In 5-fold cross-validation the influence of random noise is slightly too strong, but the results of the 100 times repeated 5-fold cross-validation are promising and encourage further research.