Albert-Ludwigs-Universität Freiburg

Bachelorarbeit

Verzerrung der Inferenz bei Verwendung gemischter Modelle in latenten Repräsentationen

Autor:
Yannick Bantel

Professor:
Prof. Dr. Harald Binder

Betreuer:
Clemens Schächter

Abgabedatum:
5. Juli 2024

universität freiburg

Inhaltsverzeichnis

1 Zusammenfassung

Eine kurze Zusammenfassung der Arbeit auf Deutsch.

Abstract

This thesis examines the potential for bias in inference when utilising mixed models in latent representations. The research primarily focuses on the application of Variational Autoencoders (VAE) combined with mixed models to analyse high-dimensional medical datasets. The study investigates the extent to which inference results may be biased and evaluates the robustness of the models used. The objective of this study is to examine the likelihood-ratio test statistics and compare them across different model configurations in order to gain insights into the reliability of using VAEs in conjunction with mixed models for statistical analysis.

1 Einleitung

Die Anwendung gemischter Modelle in latenten Repräsentationen hat in den letzten Jahren stark an Bedeutung gewonnen, besonders im Bereich des maschinellen Lernens und der Datenanalyse. Diese Modelle kombinieren feste und zufällige Effekte, um die Variabilität in den Daten besser abzubilden und somit präzisere Vorhersagen und Inferenz zu ermöglichen. Eine vielversprechende Methode zur Erstellung solcher latenten Repräsentationen ist der Variational Autoencoder (VAE). VAEs sind generative Modelle, die hochdimensionale Daten in einen niedrigdimensionalen latenten Raum transformieren, um die zugrunde liegenden Struktur zu erfassen und neue Daten zu generieren, die ähnliche Merkmale wie die Trainingsdaten aufweisen. Sie sind eine verbesserte Erweiterung der normalen Autoencoder.

Problemstellung und Motivation
Gemischte Modelle auf hochdimensionalen Datensätzen und komplexen Datenstrukturen sind oft schwer zu handhaben und mit hohen rechnerischen Leistungen verbunden. Variational Autoencoder bieten eine dimensionsreduzierte Möglichkeit Datenstrukturen in einem latenten Raum darzustellen. Die Anwendung gemischter Modelle auf einer latenten Repräsentation, wie sie der latente Raum eines Variational Autoencoder ist, würde demnach eine effizientere Berechnung und eine erleichterte Handhabung ermöglichen.
Allerdings kann die Anwendung auch zu Problemen führen. Bei der Verwendung gemischter Modelle in der latenten Repräsentation von Variational Autoencodern machen sowohl das gemischte Modell als auch der VAE Annahmen über Verteilungen über die Verteilung der Daten und der latenten Variablen. Falls die Verteilungsannahmen der Modelle verletzt werden, kommt es wiederum zu einer verzerrten Schätzung der Parameter und/oder Optimierungsgrößen. Aus diesen Gründen kann es zu einer Verzerrung der Inferenz bei Verwendung gemischter Modelle in latenten Repräsentationen kommen.

Ziel der Arbeit
Das Ziel dieser Arbeit ist es, die Verzerrung der Inferenz bei der Anwendung gemischter Modelle in latenten Repräsentationen zu untersuchen. Besonders wird analysiert, wie die Kombination von VAE und gemischten Modellen die statistischen Eigenschaften der Inferenz beeinflusst. Eine verzerrte Inferenz kann zu fehlerhaften Schlussfolgerungen führen, was die Validität der Modelle und deren praktische Anwendung beeinträchtigt.

Im ersten Teil der Arbeit werden die theoretischen Grundlagen von Variational Autoencodern und gemischten Modellen erläutert. Variational Autoencoder (VAE) ermöglichen es, die Struktur hochdimensionaler Daten zu erfassen und ähnliche neue Daten zu erzeugen. Die Architektur und das Training von VAEs werden detailliert beschrieben, um ein tiefes Verständnis ihrer Funktionsweise zu vermitteln.

Anschließend werden gemischte Modelle beschrieben, die feste und zufällige Effekte kombinieren, um die Variabilität in den Daten zu erfassen. Diese Modelle sind besonders nützlich bei der Analyse von Längsschnitt- und Cluster-Daten, wie sie in der Medizin, den Sozialwissenschaften und der Ökonomie häufig vorkommen. Ein zentrales Element der gemischten Modelle ist die Likelihood-Inferenz, bei der die Parameter durch Maximum-Likelihood-Schätzung bestimmt werden.

Im empirischen Teil der Arbeit wird ein komplexer, medizinischer Datensatz verwendet, um die Verzerrung der Inferenz bei der Anwendung gemischter Modelle auf latente Repräsentationen zu untersuchen. Die Analyse basiert auf der Likelihood-Ratio-Test Statistik, die zwischen einem vollständigen und einem reduzierten gemischten Modell unterscheidet. Durch wiederholtes Training und Evaluierung dieser Modelle auf der latenten Datenwolke des VAE wird die Verzerrung quantifiziert und bewertet.

Die Ergebnisse dieser Untersuchung tragen dazu bei, die Zuverlässigkeit von Inferenzmethoden in Kombination mit VAEs zu bewerten und liefern wertvolle Erkenntnisse für die praktische Anwendung solcher Modelle. Abschließend werden die Implikationen der Ergebnisse diskutiert und Empfehlungen für zukünftige Forschungen gegeben.

2 Theoretische Grundlagen

Im Vorfeld der Erörterung der Methodik dieser Arbeit ist eine theoretische Aufarbeitung der behandelten Themen unabdingbar. Dieses Kapitel widmet sich den theoretischen Grundlagen, die für das Verständnis und die Analyse von Verzerrungen in der Inferenz erforderlich sind, wenn gemischte Modelle in latenten Repräsentationen zum Einsatz kommen.

Das vorliegende Kapitel beginnt mit einer detaillierten Einführung in Variational Autoencoder (VAE). VAEs sind generative Modelle, die es ermöglichen, hochdimensionale Daten in niedrigdimensionale latente Repräsentationen zu überführen. Die Modelle reduzieren die Komplexität der Daten und sind in der Lage, sowohl die zugrunde liegende Struktur der Daten zu erfassen als auch neue Daten zu generieren, die ähnliche Merkmale wie die Trainingsdaten aufweisen. Die Architektur und das Training von VAEs werden detailliert beschrieben, um ein fundiertes Verständnis ihrer Funktionsweise zu vermitteln.

Im zweiten Teil des Kapitels erfolgt eine Behandlung von gemischten Modellen. Die Modelle kombinieren feste und zufällige Effekte, um die Variabilität in den Daten zu erfassen. Ihre Anwendung ist insbesondere bei der Analyse longitudinaler und Cluster-Daten von Vorteil, wie sie in den Bereichen Medizin, Sozialwissenschaften und Ökonomie häufig auftreten. Die Grundlagen gemischter Modelle, einschließlich der Annahmen und mathematischen Formulierungen, werden ausführlich erörtert.

Ein zentrales Element der theoretischen Grundlagen ist die Likelihood-Inferenz. Im Folgenden wird die Schätzung der Parameter von gemischten Modellen unter Verwendung der Maximum-Likelihood-Methode erörtert. Es wird insbesondere dargelegt, wie die Likelihood-Funktion zur Schätzung der festen und zufälligen Effekte maximiert wird und wie der Likelihood-Ratio-Test (LRT) zur Evaluierung der Modelle zum Einsatz kommt. Der Likelihood-Ratio-Test (LRT) erlaubt die Bestimmung der Signifikanz zusätzlicher Parameter sowie die Identifikation potenzieller Verzerrungen in der Inferenz.

In diesem Kapitel wird die theoretische Basis für die nachfolgenden empirischen Untersuchungen dargelegt. Der vorliegende Abschnitt bietet eine umfassende Darstellung der relevanten Methoden und Konzepte, welche erforderlich sind, um die Verzerrung der Inferenz bei der Verwendung gemischter Modelle in latenten Repräsentationen zu verstehen und zu analysieren.

2.1 Einführung in Variational Autoencoder (VAE)

Die Anwendung der gemischten Modelle auf einer latenten Repräsentation erfolgt,wie bereits erwähnt, mittels Variational Autoencoder (VAE). Variational Autoencoder sind für die Modellierung latenter Repräsentationen von großem Interesse, da sie hochdimensionale Datensätze mit Hilfe ihres Encoders im latenten Raum niedrigdimensional darstellen können. Dies reduziert die Komplexität der Modellierung und ermöglicht es, gemischte Modelle effizienter und genauer zu betreiben. Sie sind generative Modelle, welche versuchen die zugrunde liegende Struktur der Inputdaten x im latenten Raum zu modellieren.
Im Gegensatz zu herkömmlichen Autoencodern ist der VAE in der Lage, nicht nur den Eingabedatensatz zu rekonstruieren, sondern auch neue Inhalte, die ähnliche Merkmale wie die Trainingsdaten aufweisen, zu generieren. Dies wird durch die verbesserte Repräsentation ermöglicht (vgl. **bigdata-insider-vae**). Insbesondere wird der latente Raum nicht wie bei normalen Autoencoder durch feste Punkte modellieret, wie es

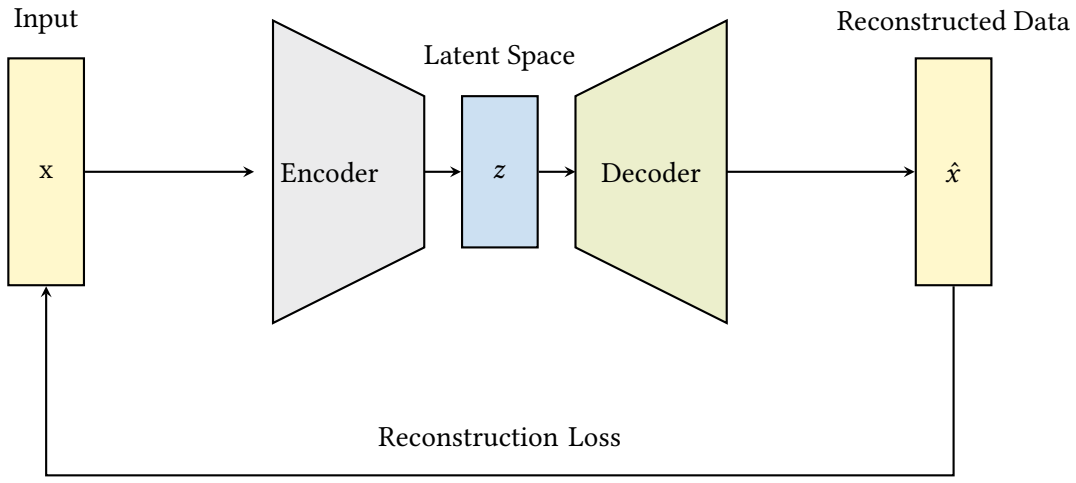


Abbildung 2.1 Aufbau eines herkömmlichen Autoencoders

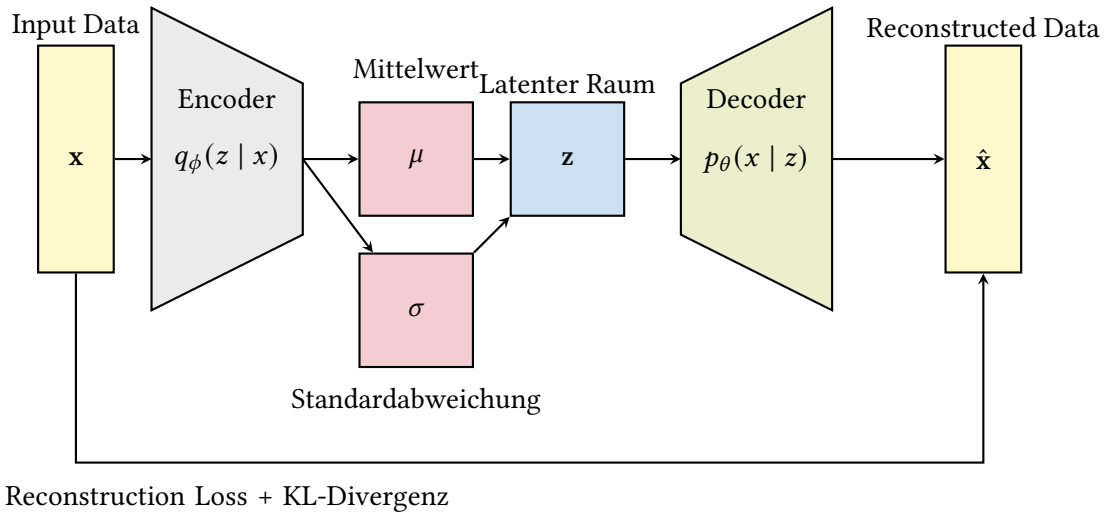


Abbildung 2.2 Darstellung der Architektur eines Variational Autoencoders (VAE)

in Abbildung 2.2 dargestellt ist, sondern in der Erweiterung VAE durch eine Wahrscheinlichkeitsverteilung (Normalverteilung).

2.1.1 Strucktur des VAEs

Die Architektur eines VAE basiert auf zwei neuronalen Netzwerken (einem Encoder Modell und einem Decoder Modell). Der Encoder ist ein neuroanles Netzwerk, das die hochdimensionalen Eingabedaten x durch mehrere Schichten hindurch in eine niedrigdimensionale latente Repräsentation z transformiert. Diese Transformation ermöglicht es, die zugrunde liegende Struktur der Daten zu erfassen und in einer komprimierten Form darzustellen.
Die latenten Variablen werden als Verteilung in Form eines Mittelwerts μ und einer Standardabweichung σ kodiert. Der Decoder transformiert die latenten Daten so genau wie möglich zurück in die ursprünglichen Eingabedaten. Dies ermöglicht es, neue Datenpunkte zu generieren, die ähnliche Eigenschaften wie die Trainingsdaten aufweisen. Beide Modelle bestehen jeweils aus mehreren neuronalen Schichten, die jeweils die Transformation durchführen und lernen die wesentlichen Merkmale der Eingabedaten zu extrahieren und eine komprimierte Version dieser Daten zu erzeugen.

Latenter Raum

Variablen, die man nicht direkt messen kann, demnach nicht Teil des erhaltenen Datensatzes sind, bezeichnet man als latente Variablen. Sie werden erst mithilfe der gegebenen Daten erschlossen und ergeben im Verbund den latenten Raum.
Im VAE werden die latenten Variablen z aus der prior-Verteilung $p(z)$ gezogen, welche eine multivariate Normalverteilung $p(z) = \mathcal{N}(0, I)$ ist. Die latenten Daten sind der Output aus dem Encoder Modell, welches die approximierte posterior Verteilung $q_\phi(z|x)$ parametrisiert, wobei ϕ der Parametervektor des Encoders ist. Dieser erlernt somit zwei Vektoren, nämlich den Mittelwert μ_ϕ und die Standardabweichung σ_ϕ der Normalverteilung $q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \sigma^2_\phi(x))$.
Der Decoder versucht aus den latenten Variablen die Inputdaten x durch die Likelihood-Verteilung $p_\theta(x|z)$ zu rekonstruieren. Die Likelihood-Verteilung der Daten gegeben die latenten Daten wird durch den Parametervektor θ des Decoders parametrisiert.
Die Wahrscheinlichkeit, dass die beobachteten Daten aus den latenten Repräsentationen generiert wurden, wird durch dieses Decoder-Modell modelliert. Auch hier wird typischerweise eine Normalverteilung angenommen, sofern die Daten reellwertig sind. Im Falle binärer Daten wird die Verteilung als Bernoulli-Verteilung modelliert.
Für weiterführende Details wird auf die Publikation **Auto-EncodingVariationalBayes** verwiesen.

2.1.2 Training VAE

Das Training eines Variational Autoencoder basiert auf den Prinzipien der Variationsinferenz, einer Methode zur Approximation komplexer posterior Verteilungen. Die Berechnung der posterior Verteilung ist besonders bei komplexen Modellen mit Schwierigkeiten verbunden. Infolgedessen wird bei der Variationsinferenz eine einfachere Verteilung verwendet, um die wahre posteriore Verteilung zu approximieren, wodurch Berechnungen effizienter und skalierbarer werden. Im Kontext eines VAE wird die wahre posterior Verteilung $p(z|x)$ der latenten Variablen in Abhängigkeit der Input-Daten, durch eine einfachere Verteilung $q_\phi(z|x)$ approximiert. Das Ziel ist es die Parameter dieser Verteilung so zu optimieren, dass $q_\phi(z|x)$ so nah wie möglich an $p(z|x)$ liegt. Diese Annäherung wird durch die Maximierung des Evidence Lower Bound (ELBO) erreicht, der eine untere Schranke der Datenloglikelihood darstellt.
Zur Effizienten Berechnung wird der Reparametrisierungs Trick verwendet, welcher die Anwendung von Gradientenverfahren zur Optimierung der Modellparameter ermöglicht.

In diesem Abschnitt werden die wesentlichen mathematischen Aspekte hinter dem Training des VAE erläutert und beschrieben.

Das Training eines Variational Autoencoders (VAE) umfasst mehrere Schritte, deren Ziel es ist, die Parameter des Modells so anzupassen, dass der Evidence Lower Bound (ELBO) maximiert wird. Der Prozess lässt sich in drei Hauptkomponenten unterteilen: die Definition des ELBO, die Anwendung des Reparameterization Trick und die Optimierung des Modells mittels stochastischer Gradientenverfahren.
Eine im Rahmen dieser Arbeit und auch im Bereich Machine Learning wichtige Verteilung ist die Normal- oder auch Gauß-Verteilung. Diese wird als erste Definition in dieser Arbeit eingeführt:

Definition 2.1.1 (Normal-/Gaußverteilung)**.**
Seien $\mu, \sigma \in \mathbb{R}$ mit $\sigma > 0$. Die Zufallsvariable X ist normalverteilt mit Erwartungswert μ und Standardabweichung σ bzw. Varianz σ^2, falls X die folgende Wahrscheinlichkeitsdichte hat:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

Notation: $X \sim \mathcal{N}(\mu, \sigma^2)$

Ein wichtiges Beispiel der Normalverteilung ist die Standardnormalverteilung, welche eine Normalverteilung mit den Parametern $\mu = 0$ und $\sigma^2 = 1$ ist ($X \sim \mathcal{N}(0, 1)$). Eine solche Standardnormalverteilung hat

die Dichtefunktion

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}.$$

Der ELBO lässt sich aus dem Rekonstruktionsverlust (Reconstruction Loss) und der Kullback-Leibler-Divergenz zusammensetzten. Die Kullback-Leibler-Divergenz quantifiziert die Differenz zwischen der approximierten posterior Verteilung und der prior Verteilung und ist folgendermaßen definiert:

Definition 2.1.2 (Kullback-Leibler-Divergenz (KL-Divergenz))**.**
Seien Q und P zwei Wahrscheinlichkeitsverteilungen. Sei dabei P die wahre Verteilung mit Dichtefunktion p(x) und Q die approximierte Verteilung mit Dichtefunktion q(x). Dann ist die KL-Divergenz zwischen Q und P definiert als

$$D_{KL}(P||Q) = \int p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

Der Rekonstruktionsverlust misst, wie gut der Decoder die Input Daten rekonstruiert hat. Er wird in einem Variational Autoencoder als negativer log-Likelihood der Daten x gegeben die latenten Daten z angegeben. Er ist durch

$$\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$$

gegeben. Betrachtet man nun die konstante log-Likelihood der Daten unabhängig von z, so kann man aus ihr den ELBO herleiten (vgl. **IntroductiontoVAEs**):

$$\begin{aligned}
\log p_\theta(x) &= \log p_\theta(x) * \overbrace{\int q_\phi(z|x)dz}^{=1} && (2.1)\\
&= \int \log p_\theta(x) q_\phi(z|x) dz && (2.2)\\
&= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x)] && (2.3)\\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log\left[\frac{p_\theta(x,z)}{p_\theta(z|x)}\right]\right] && (2.4)\\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log\left[\frac{p_\theta(x,z) q_\phi(z|x)}{q_\phi(z|x) p_\theta(z|x)}\right]\right] && (2.5)\\
&= \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[\log\left[\frac{p_\theta(x,z)}{q_\phi(z|x)}\right]\right]}_{\substack{=\mathcal{L}(\theta,\phi;x)\\ \text{(ELBO)}}} + \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[\log\left[\frac{q_\phi(z|x)}{p_\theta(z|x)}\right]\right]}_{=D_{KL}(q_\phi(z|x)||p_\theta(z|x))} && (2.6)
\end{aligned}$$

Der zweite Term in Gleichung 2.6 ist nach Definition die nicht negative Kullback-Leibler-Divergenz (KL-Divergenz) zwischen $q_\phi(z|x)$ und $p_\theta(z|x))$ und der erste Term in Gleichung 2.6 stellt den Evidence Lower Bound (ELBO) dar.
Dieser wird wir folgt definiert:

Definition 2.1.3 (Evidence Lower Bound (ELBO) für VAEs)**.**
Sei z die latente Zufallsvariable und seien x die Input Daten. Sei $q_\phi(z|x)$ das Verteiilung von z gegeben x und $p_\theta(x,z)$ die gemeinsame Verteilung von z und x. Der ELBO ist definiert durch

$$\mathcal{L}(\theta,\phi;x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x,z) - \log q_\phi(z|x)]$$

Die ELBO kann auch durch den Rekonstruktionsfehler und der KL-Divergenz definiert. Die ELBO zielt darauf ab, die log-Likelihood $\log p_\theta(x)$ zu maximieren. Es ist leicht zu sehen, dass durch Umstellen der Gleichung 2.6 die ELBO, da die KL-Divergenz nicht negativ ist, eine untere Schranke der Log-Likelihood bietet (Vgl. Gleichung 2.7).

$$\mathcal{L}(\theta, \phi; x) = \log p_\theta(x) - D_{KL}(q_\phi(z|x)||p_\theta(z|x)) \tag{2.7}$$
$$\leq \log p_\theta(x) \tag{2.8}$$

Alternativ kann dies mit der Jensenschen Ungleichung (vgl. **JensenscheUngleichung**) hergeleitet werden:

$$\log p_\theta(x) = \log \int p_\theta(x, z) dz \tag{2.9}$$
$$= \log \int p_\theta(x, z) \frac{q_\phi(z|x)}{q_\phi(z|x)} dz \tag{2.10}$$
$$= \log \mathbb{E}_{q_\phi(z|x)} \left[\frac{p_\theta(x, z)}{q_\phi(z|x)} \right] \tag{2.11}$$
$$\overset{Jensen}{\geq} \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p_\theta(x, z)}{q_\phi(z|x)} \right] = \mathcal{L}(\theta, \phi; x) \tag{2.12}$$

Es ist sofort ersichtlich, dass die log-likelihood $p_\theta(x)$ durch Maximieren der ELBO bzgl. θ und ϕ selbst maximiert wird und somit die Qualität unseres generatives Modells verbessert wird. Gleichzeitig minimiert sich dadurch die KL-Differenz zwischen der approximativen Verteilung $q_\phi(z|x)$ und den wahren posterior Verteilung $p_\theta(z|x)$. Durch die Maximierung der ELBO wird also die Approximation $q_\phi(z|x)$ an die posterior Verteilung optimiert.

Die Maximierung der ELBO kann durch stochastische Gradientenverfahren wie Stochastic Gradient Descent (SGD) oder andere fortschrittliche Verfahren erfolgen. Die Berechnung der Gradienten des ELBOs bzgl. θ stellt keine Probleme dar. Mit dem für solche Methoden üblichen Monte-Carlo Schätzer (Gleichung 2.16) lassen sich die Gradienten bzgl. θ einfach berechnen, wie man in den folgenden Gleichungen sehen kann.

$$\nabla_\theta \mathcal{L}(\phi, \theta; x) = \nabla_\theta \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x, z) - \log q_\phi(z|x)] \tag{2.13}$$
$$= \mathbb{E}_{q_\phi(z|x)} [\nabla_\theta (\log p_\theta(x, z) - \log q_\phi(z|x))] \tag{2.14}$$
$$\approx \nabla_\theta (\log p_\theta(x, z) - \log q_\phi(z|x)) \tag{2.15}$$
$$= \nabla_\theta \log p_\theta(x, z) \tag{2.16}$$

Der Monte-Carlo-Schätzer für Gradienten ist eine gängige Methode, die verwendet wird um die notwendigen Gradienten zu berechnen, die zur Optimierung der Variationsparameter ϕ führen. Der Erwartungswert einer Funktion $f(x)$ unter einer Wahrscheinlichkeitsverteilung $p_\theta(x)$ kann mittels Monte-Carlo Schätzung wie folgt angenähert werden (vgl. **MonteCarloEstimation**):

$$\mathbb{E}_{p_\theta(x)}[f(x)] \approx \frac{1}{N} \sum_{n=1}^{N} f(\hat{x}^{(n)}), \tag{2.17}$$

wobei $\hat{x^{(n)}}$ unabhängige Stichproben sind, die aus der Verteilung $p_\theta(x)$ gezogen wurden.
Im Falle der Variationsinferenz im VAE kann der Gradient des Erwartungswert mit dem Monte-Carlo Schätzer approximiert werden.

$$\nabla_\phi \mathbb{E}_{q_\phi(z|x)}[f(z)] = \mathbb{E}_{q_\phi(z|x)}[\nabla_\phi f(z)] \approx \frac{1}{N} \sum_{n=1}^{N} \nabla_{q_\phi} f(z^{(n)})) \tag{2.18}$$

wobei $z^{(n)} \sim q_\phi(z|x)$ ist.
Allerdings ist die Berechnung der Gradienten von $\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x,z) - \log q_\phi(z|x)\right]$ bezüglich des Variationsparameters ϕ problematisch, da der Erwartungswert des ELBO bzgl. $q_\phi(z|x)$ genommen wird und die Funktion $q_\phi(z|x)$ von ϕ abhängt.

$$\nabla_\phi \mathcal{L}(\theta,\phi;x) = \nabla_\phi \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x,z) - \log q_\phi(z|x)\right] \tag{2.19}$$
$$\neq \mathbb{E}_{q_\phi(z|x)}\left[\nabla_\phi(\log p_\theta(x,z) - \log q_\phi(z|x))\right] \tag{2.20}$$

Zur Lösung dieses Problems wird der sogenannte Reparameterization-Trick eingesetzt, welcher die Zufallsvariable transformiert, um die Gradienten-Berechnung zu vereinfachen.

Reparametrisierungs Trick

Der Reparameterisierungs-Trick ist eine Methode zur Vereinfachung der Gradientenberechnung in Variational-Autoencodern. Er ermöglicht eine effizientere Berechnung der Gradienten der Evidence Lower Bound und somit eine effizientere Optimierung dessen. Der Reparameterization Trick transformiert die Zufallsvariable z in eine andere von z unabhängige deterministische Funktion von einer in eine von z unabhängigen Hilfsvariablen ϵ. Sei also die latente Variable z, die aus $q_\phi(z|x)$ gezogen wurde, gegeben. Sie wird nun als deterministische Funktion einer Hilfsvariablen ϵ unabhängig von ϕ ausgedrückt. Die Transformation sieht dann wie folgt aus:

$$z = g(\epsilon, x, \phi)$$

Dabei ist $g(\epsilon, x, \phi)$ eine differenzierbare Funktion und ϵ eine Zufallsvariable mit einer bekannten Verteilung (z.B. $\epsilon \sim \mathcal{N}(0, I)$).
Im Falle einer Gaußverteilung $z \sim \mathcal{N}(\mu, \sigma^2)$ könnte die Umparametrisierung wie folgt aussehen

$$z = \mu + \sigma \odot \epsilon \quad \text{mit } \epsilon \sim N(0, I).$$

Dabei sind μ und σ die Inferenzparameter ϕ. Durch die Umparametrisierung können die Gradienten bezüglich ϕ effizient berechnet werden, da der Erwartungswert über $q_\phi(z|x)$ sich nun als Erwartungswert über $p(\epsilon)$ ersetzen lässt (vgl. **MonteCarloEstimation**).

$$\nabla_\phi \mathbb{E}_{q_\phi(z)}[f(z)] = \nabla_\phi \int q_\phi(z) f(z) dz \tag{2.21}$$
$$= \nabla_\phi \int p(\epsilon) f(g_(\epsilon, x, \phi)) d\epsilon \tag{2.22}$$
$$= \nabla_\phi \mathbb{E}_{p(\epsilon)}[f(g(\epsilon, x, \phi))] \tag{2.23}$$
$$= \mathbb{E}_{p(\epsilon)}[\nabla_\phi f(g(\epsilon, x, \phi))] \tag{2.24}$$

Die Erwartung des ELBO lässt sich demnach ebenso umschreiben zu:

$$\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x,z) - \log q_\phi(z|x)\right] = \mathbb{E}_{p(\epsilon)}\left[\log p_\theta(x, g(\epsilon,x,\phi)) - \log q_\phi(g(\epsilon,x,\phi)|x)\right]$$

Der Reparameterisierungstrick bietet somit eine effiziente und flexible Methode zur Berechnung von Gradienten in Modellen mit latenten Variablen und ermöglicht die Anwendung leistungsstarker Optimierungsmethoden wie SGD auf komplexe probabilistische Modelle. Wie der Reparametrisierungstrick in einem VAE aussieht ist in Abbildung 2.3 veranschaulicht.
Mit der neuen Darstellung kann der Gradient des ELBO berechnet werden als:

$$\nabla_\phi \mathcal{L}(\theta,\phi;x) = \mathbb{E}_{p(\epsilon)}\left[\nabla_\phi(\log p_\theta(x, g(\epsilon,x,\phi)) - \log q_\phi(g(\epsilon,x,\phi)|x))\right]$$

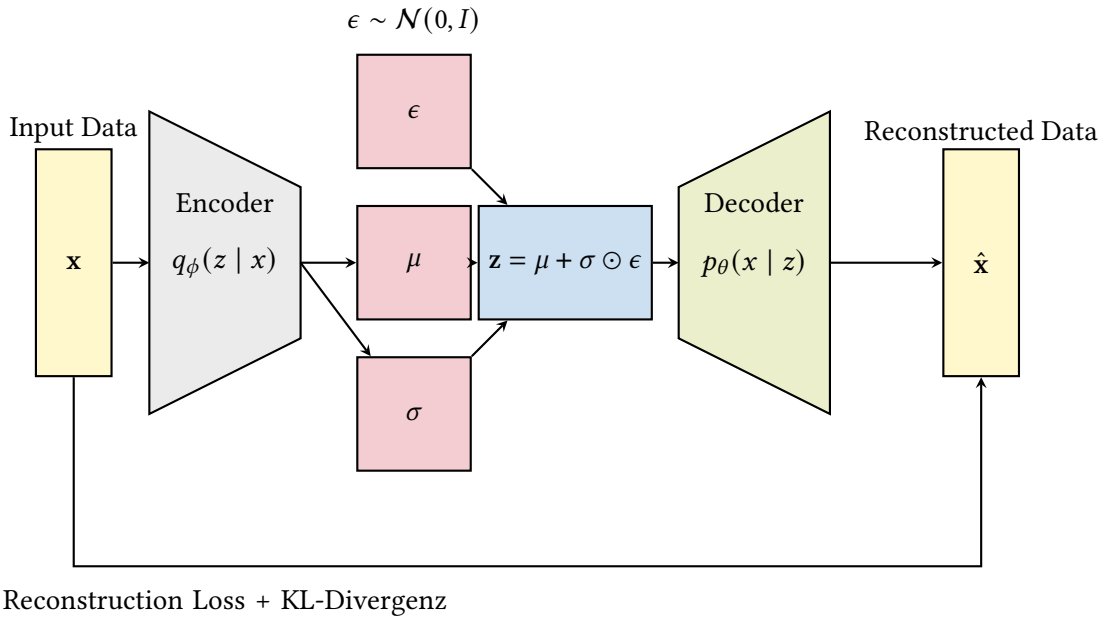


Abbildung 2.3 Architektur eines VAEs mit Reparameterization Trick

mit $z = g(\epsilon, x, \phi)$. Die Erwartung auf der rechten Seite wird durch Monte-Carlo-Sampling approximiert, indem man mehrere Stichproben von ϵ zieht, die entsprechende Transformation anwendet und dann den Durchschnitt der Gradienten bildet:

$$\nabla_\phi \mathcal{L}(\theta, \phi; x) \approx \frac{1}{N} \sum_{n=1}^{N} \nabla_\phi (\log p_\theta(x, g(\epsilon_n, x, \phi)) - \log q_\phi(g(\epsilon_n, x, \phi)|x))$$

Hier sind ϵ_n die unabhängigen Stichproben aus der Verteilung p(ϵ) (vgl. **MonteCarloEstimation**).

2.2 Grundlagen Gemischte Modelle

Das Ziel der Arbeit ist es die Anwendung gemischter Modelle in latenten Repräsentationen auf eine mögliche Verzerrung zu untersuchen. Die nötige Theorie der latenten Repräsentation ist nun in Form eines VAE gegeben. Im folgenden Kapitel werden nun die mathematischen Grundlagen zu gemischten Modellen eingeführt.

Ein gemischtes Modell stellt ein statistisches Verfahren zur Datenanalyse dar, welches sowohl feste als auch zufällige Effekte (fixed and random effects) modelliert. Gemischte Modelle finden insbesondere bei der Analyse von longitudinalen und Cluster spezifischen Daten, welche aus zeitlich wiederholten Beobachtungen (y_{it}, x_{it}), $t = 1, ..., T_i$ für jedes Individuum $i = 1, ..., n$ bestehen, ihre Anwendung. Die Variable y kennzeichnet dabei eine Antwortvariable, während x einen Vektor von Kovariablen darstellt. Ein Beispiel für einen solchen Datensatz könnte ein medizinischer Datensatz sein,

$$(y_i, x_i) = (y_{i1}, ..., y_{iT_i}, x_{i1}, ..., x_{iT_i})$$

bei dem y_{ij} eine Beobachtung an Individuum i zum Zeitpunkt t_{ij} bezeichnet und T_i die Anzahl an Beobachtungen ist.

Zur Einführung der gemischten Modelle folgen wir den Notationen in **fahrmeir-2001-multivariate** und **fahrmeir-2011-regression**. Longitudinal und Cluster spezifische Daten weisen zwei Ebenen auf. Im Folgenden betrachten wir das Beispiel des oben eingeführten medizinischen Datensatzes. Die erste Ebene bezieht sich dabei auf die Daten innerhalb einer Gruppe oder eines Individuums. In diesem Fall umfasst

die erste Ebene den Patienten als Individuum mit seinen unterschiedlichen Werten für die Tests entlang der Zeitreihe T_i. Auf der allgemeineren zweiten Ebene erfolgt eine Betrachtung aller Patienten.

Im Rahmen eines gemischten Modells wird auf der ersten Ebene angenommen, dass die Antwortvariablen linear von den unbekannten bevölkerungsspezifischen festen Effekten β und den unbekannten Cluster spezifischen zufälligen Effekten b_i abhängen.
Die folgende Gleichung beschreibt ein gemischtes Modell für ein Individuum i zum Zeitpunkt t:

$$y_{it} = x_{it}^t \beta + w_{it}^t b_i + \epsilon_{it} \tag{2.25}$$

Innerhalb des Modells werden die Designvektoren x_{it} und w_{it} als unabhängige Variablen definiert, wobei x_{it} beispielsweise die Testwerte in einem medizinischen Datensatz repräsentiert. Die Zufallsvariable ϵ_{it} hingegen ist unkorreliert und folgt einer normalverteilten Wahrscheinlichkeitsdichte mit Erwartungswert $\mathbb{E}(\epsilon_{it}) = 0$ und Varianz $Var(\epsilon_{it}) = \sigma^2$. Der Ausdruck a^t bezeichnet den transponierten Vektor, bzw. die transponierte Matrix von a.

Betrachtet man nun die zweite Ebene, so werden die zufälligen Effekte b_i zwischen den verschiedenen Individuen gemäß einer Mischverteilung mit Erwartungswert $\mathbb{E}(b_i) = 0$ unabhängig variieren. Es wird angenommen, dass die zufälligen Effekte b_i unabhängig und identisch normalverteilt sind.

$$b_i \sim \mathcal{N}(0, Q) \tag{2.26}$$

Dabei ist $Cov(b_i) = Q > 0$ die (q x q) Kovarianzmatrix, welche symmetrisch und positiv semi-definit ist. Die Größe q beschreibt dabei die Anzahl der zufälligen Effekte. Eine ausführliche Beschreibung findet sich in **pinheiro2000** (Kapitel 2.2.1).
Aufgrund dieser Überlegungen lässt sich nun das Model 2.25 in eine allgemeinere Form bringen:

Definition 2.2.1 (Lineares gemischtes Modell für Longitudinal- oder Clusterdaten)**.**
Seien $X_i = (x_{i1}, ..., x_{iT_i})$ und $W_i = (w_{i1}, ..., w_{iT_i})$ bekannte Designmatrizen für die festen und zufälligen Effekte. Seien β ein p-dimensionaler Vektor von festen Effekten und b_i ein q-dimensionaler Vektor von zufälligen Effekten und sei $\epsilon_i = (\epsilon_{i1}, ..., \epsilon_{iT_i})$ der normalverteilte Fehlervektor.

Ein lineares gemischtes Modell für den T_i-dimensionalen Antwortvektor der i-ten Gruppe wird durch

$$y_i = X_i * \beta + W_i * b_i + \epsilon_i$$
$$b_i \sim \mathcal{N}(0, Q), \epsilon_i \sim \mathcal{N}(0, R = \sigma_\epsilon^2 I)$$

definiert.

Die Daten der zufälligen und festen Effekte werden in den Designmatrizen (oder Datenmatrizen) X_i und W_i gespeichert. Die Parametervektoren β (für die festen Effekte) und b_i (für die zufälligen Effekte) initialisieren den Einfluss der Daten auf den Antwortvektor. Um auch für immer auftretende Messfehler oder unerwartete Einflüsse gewappnet zu sein, wird ein zufälliges Rauschen ϵ hinzugefügt.

Aufgrund des normalverteilten Fehlervektors kann nun auch ein marginales Modell als multivariates heteroskedastisches lineares Regressionsmodell definiert werden. Dieses Modell ist für die Berechnung der Likelihood-Inferenz von entscheidender Bedeutung.

Definition 2.2.2 (Marginales gemischtes Modell)**.**
Seien die Annahmen von 2.2.1 gegeben. Das marginale gemischte Modell ist definiert als

$$y_i = X_i \beta + \epsilon_i^*,$$

mit dem multivariaten Fehlervektor $\epsilon_i^* = (\epsilon_{i1}^*, ..., \epsilon_{iT_i}^*)$ mit $\epsilon_{it}^* = w_{it}^T b_i + \epsilon_i$. Die ϵ_{it}^* sind dabei unabhängig und identisch verteilt (i.i.d.),

$$\epsilon_i^* \sim \mathcal{N}(0, V_i), \quad mit \quad V_i = \sigma_\epsilon^2 I + W_i Q W_i^t \tag{2.27}$$

Letztendlich können die einzelnen Cluster/Gruppen zu einem einzigen allgemeinen linearen gemischten Modell zusammengefasst werden, welches wie folgt definiert wird:

Definition 2.2.3 (Allgemeines lineares gemischtes Modell)**.**
Ein lineares gemischtes Modell ist definiert durch

$$y = X\beta + Wb + \epsilon$$

mit

$$\begin{pmatrix} b \\ \epsilon \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} Q & 0 \\ 0 & R = \sigma_\epsilon^2 I \end{pmatrix}\right)$$

gegeben. Dabei sind X, bzw W die Designmatrizen der festen, bzw zufälligen Effekte, β und b die Parametervektoren der festen und der zufälligen Effekten und ϵ der Fehlervektor. Insbesondere sind Q und R die Kovarianzmatrizen der von b und ϵ.

R ist also die Kovarianzmatrix des Fehlervektors ϵ. Sie beschreibt die Varianz des Fehlers und eventuelle Korrelationen zwischen den Fehlern. Die Kovarianzmatrix der zufälligen Effekte b ist durch Q gegeben. Sie beschreibt die Varianz und die Korrelationen der zufälligen Effekte über die verschiedenen Gruppen oder Individuen. In Konsequenz der Definition 2.2.3 lässt sich das marginale Modell 2.2.2 verallgemeinern zu:

$$y = X\beta + \epsilon^* \tag{2.28}$$

wobei $\epsilon^* = Wb + \epsilon$ ist mit $\epsilon^* \sim \mathcal{N}(0, V)$ und der Gesamtkovarianzmatrix $V = R + WQW^t$. Die Gleichung 2.28 beschreibt also das allgemeine Marginale gemischte Modell.

2.3 Likelihood Inferenz und Verzerrung

Um die Verzerrung der Inferenz messen zu können, ist es zunächst erforderlich, die Theorie zur Likelihood-Inferenz von gemischten Modellen einzuführen. Dies umfasst sowohl die Schätzung der Parameter der zufälligen Effekte b_i als auch die Schätzung der Parameter β, σ_ϵ und Q. Um die Verzerrung zu quantifizieren, wird der sogenannte Likelihood-Ratio Test (LRT) eingeführt, welcher hilft den Einfluss eines zusätzlichen Effekts in gemischten Modellen zu messen. Wie dieser Test genau funktioniert und wie der Likelihood-Ratio Test durchgeführt wird, werden wir später erläutern. Zuvor benötigen wir noch etwas Theorie zur Likelihood-Berechnung.

2.3.1 Likelihood Berechnung gemischter Modelle

Im Folgenden wird die Schätzung der unbekannten Parameter erörtert. Der Vorliegende Ansatz basiert auf den Ausführungen von **fahrmeir-2011-regression**.

Die Berechnung der Schätzer erfolgt mittels der Maximum-Likelihood Methode. Als Alternative kann die restringierte ML-Methode heran gezogen werden, die jedoch nicht für den Likelihood-Ratio Test geeignet ist. Daher wird die Berechnung der Parameter bei der ML-Methode belassen.
Die Schätzung der Parameter in einem gemischten Modell ist jedoch mit gewissen Schwierigkeiten verbunden. Neben dem β sind auch b_i, Q und σ_ϵ unbekannt. Daher ist es erforderlich, sowohl die festen und zufälligen Effekte als auch die unbekannten Parameter in Q und σ_ϵ, die wir als δ bezeichnen, zu schätzen. Dies bedingt eine geschachtelte Schätzung.

Im Folgenden wird zunächst angenommen, dass die Kovarianzen R, bzw. σ_ϵ, und Q bekannt sind. In diesem Zusammenhang ist auch V gemäß 2.28 bekannt. Für die Schätzung von β, ausgehend vom marginalen Modell, bietet sich

$$\hat{\beta} = (X^t V^{-1} X)^{-1} X^t V^{-1} y \tag{2.29}$$

an. Dieser Kleinste-Quadrate-Schätzer (KQ-Schätzer) für β ergibt sich aus dem verallgemeinertem Kleinste-Quadrate Kriterium (vgl. **KQ-SchÃďtzer**), welches die quadratische Verlustfunktion unter Berücksichtigung von V

$$L(\beta) = (y - X\beta)^t V^{-1}(y - X\beta)$$

bezüglich β minimiert. Siehe hierzu auch **fahrmeir-2011-regression** (Kap. 3).
Der KQ-Schätzer ist gleichzeitig der log-Likelihood Schätzer unter der Normalverteilungsannahme. Dazu wird zuerst die Log-Likelihood-Funktion definiert, welche sich aus der Likelihood-Funktion von y gegeben β und δ herleiten lässt:

$$L(\beta, \delta|y) = \frac{1}{(2 * \pi)^{\frac{n}{2}} |V|^{\frac{1}{2}}} * \exp(-\frac{1}{2}(y - X\beta)^t V^{-1}(y - X\beta)) \tag{2.30}$$

$$\tag{2.31}$$

Wendet man den Logarithmus auf die Likelihood an und vereinfacht diesen Term erhält man die Log-Likelihood:

$$l(\beta, \delta|y) = \log L(\beta, \delta|y) = \log(\frac{1}{(2 * pi)^{\frac{n}{2}} |V|^{\frac{1}{2}}}) - \frac{1}{2}(y - X\beta)^t V^{-1}(y - X\beta) \tag{2.32}$$

$$= -0.5 * (log(|V|) + (y - X\beta)^t V^{-1}(y - X\beta) + N * log(2\pi)) \tag{2.33}$$

Nach dieser Herleitung folgt die Definition:

Definition 2.3.1 (Log-Likelihood-Funktion)**.**
Sei $y = X\beta + \epsilon^*$ ein marginales Modell, wie in 2.28 gegeben und sei δ der Parametervektor der Kovarianzmatrix V. Die Log-Likelihood-Funktion der Daten y gegeben β und δ is definiert durch

$$l(\beta, \delta|y) = -0.5 * (log(|V|) + (y - X\beta)^t V^{-1}(y - X\beta) + N * log(2\pi))$$

Dabei ist N die Anzahl der Beobachtungen des marginalen Modells und $|V|$ die Determinante von V.

Ableiten der Log-Likelihood von β nach β ergibt den KQ-Schätzer aus 2.29.

$$\frac{d}{d\beta} l(\beta) = X^t V^{-1}(y - X\beta) \stackrel{!}{=} 0 \Rightarrow \hat{\beta} = (X^t V^{-1} X)^{-1} X^t V^{-1} y$$

Siehe hierzu auch **fahrmeir-2011-regression** (Kap. 3).

Gemäß dem Gauß-Markov-Theorem stellt $\hat{\beta}$ den besten linearen erwartungstreuen Schätzer (BLUE, best linear unbiased estimator) für die festen Effekte dar (vgl. **Statistik-Wiwi**). Zur Ermittlung des Schätzers ist lediglich eine Schätzung der Parameter δ in V sowie der Einsatz des Schätzers $\hat{V}$ von V in $\hat{\beta}$ erforderlich.
Für den Schätzer von b verwenden wir den bedingten Erwartungswert $E(b|y)$ von b gegeben die Daten y, welcher unter der Normalverteilungsannahme der beste Schätzer ist (vgl. **fahrmeir-2011-regression** Kap. 6.3.1).
Im Folgenden wird nun die gemeinsame Verteilung von b und y betrachtet, welche folgendermaßen dargestellt wird:

$$\begin{pmatrix} y \\ b \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} X\beta \\ 0 \end{pmatrix}, \begin{pmatrix} V & WQ \\ QW^t & Q \end{pmatrix}\right)$$

In Anbetracht dessen erhalten wir $E(b|y) = QW^t V^{-1}(y - X\beta)$.
Ersetzt man nun β durch den Schätzer $\hat{\beta}$ erhält man den Schätzer für die zufälligen Effekte

$$\hat{b} = \hat{Q} W^t \hat{V}^{-1}(y - X\hat{\beta}).$$

Dieser ist der beste lineare unverzerrte Schätzer (BLUP, best linear unbiased prediction).
Die Schätzer für die festen und zufälligen Effekte lassen sich also folgendermaßen definieren:

Definition 2.3.2 (Schätzer für feste und zufällige Effekte)**.**
Sei $y = X\beta + Wb + \epsilon$ ein lineares gemischtes Modell und $y = X\beta + \epsilon^*$ das zugehörige Marginale nach 2.28. Dann ist

$$\hat{\beta} = (X^t \hat{V}^{-1} X)^{-1} X^t \hat{V}^{-1} y$$

ein Schätzer für die festen Effekte und

$$\hat{b} = \hat{Q} Z^t \hat{V}^{-1} (y - X\hat{\beta})$$

ein Schätzer für die zufälligen Effekte.

Wie man den Schätzer der zufälligen und festen Effekte erhält im Falle, dass die Kovarianzen bekannt sind, wurde bereits gezeigt. Nun gilt es noch den die Berechnung des Kovarianzschätzer einzuführen, damit die Schätzer der zufälligen und festen Effekte tatsächlich berechnet werden können.
Wie bereits erwähnt, soll der Parametervektor δ alle unbekannten Parameter in den Kovarianzen V, Q und σ_ϵ enthalten. Anhand des Schätzers $\hat{\delta}$ lassen sich dann der Kovarianzschätzer sowie die Schätzer der festen und zufälligen Effekte berechnen.
Die ML Schätzung für δ basiert auf dem marginalen Modell

$$y \sim \mathcal{N}(X\beta, V).$$

Es wird im Folgenden die Log-Likelihood von β und δ abzüglich des Konstanten Terms betrachtet:

$$l(\beta, \delta) = -\frac{1}{2}(log(|V|) + (y - X\beta)^t V^{-1} (y - X\beta))$$

Maximiert man diese bezüglich β für festes δ, erhält man folgenden Schätzer:

$$\hat{\beta} = (X^t V^{-1} X)^{-1} X^t V^{-1} y.$$

Setzt man nun $\hat{\delta}$ in $l(\beta, \delta)$ ein, so erhält man die Profil-Log-Wahrscheinlichkeit

$$l(\delta)_p = -\frac{1}{2}(log(|V|) + (y - X\hat{\beta})^t V^{-1} (y - X\hat{\beta})).$$

Folglich erhält man den ML-Schätzer $\hat{\delta}_{ML}$ durch Maximierung von $l(\delta)_p$, welcher wie folgt definiert wird:

Definition 2.3.3 (Kovarianz-Schätzer)**.**
Sei $y = X\beta + Wb + \epsilon$ ein lineares gemischtes Modell mit

$$\begin{pmatrix} b \\ \epsilon \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} Q & 0 \\ 0 & R \end{pmatrix} \right)$$

und sei δ der unbekannte Parametervektor von Q,R und $V = Var(y)$.
Dann ist $\hat{\delta}_{ML}$ der ML-Schätzer für δ, den man durch maximieren von

$$l(\delta)_p = -\frac{1}{2}(log(|V|) + (y - X\hat{\beta})^t V^{-1} (y - X\hat{\beta}))$$

erhält. Dabei ist

$$\hat{\beta} = (X^t V^{-1} X)^{-1} X^t V^{-1} y$$

Mit dem Schätzer $\hat{V}$ lassen sich die Schätzer der festen und zufälligen Effekte nun berechnen.
Um die Verzerrung der Inferenz messen zu können, müssen wir die Log-Likelihood Werte berechnen können. Diese werden dann mit dem Likelihood-Ratio-Test ausgewertet. Der Log-Likelihood Wert eines gemischten Modell ergibt sich aus der Maximum-Likelihood (ML)-Methode und ist folgendermaßen definiert:

Definition 2.3.4 (log-Likelihood Wert für ein gemischtes Modell)**.**
Sei $y = X\beta + \epsilon^*$ wie in 2.28 definiert mit $V = R + WQW^t$ und N die Anzahl der Daten in der Designmatrix W. Sei $r = y - X(X^tV^{-1}X)^{-1}X^tV^{-1}y$ und p der Rang von X. Dann ist die Log-Likelihood definiert als:

$$l_{ML}(Q, R) = -0.5 * (log(|V|) + r^tV^{-1}r + N * log(2\pi))$$

Und die Restricted-Log-Likelihood ist definiert durch:

$$l_{REML}(Q, R) = -0.5 * (log(|V|) + X^tV^{-1}X + r^tV^{-1}r + (N - p) * log(2\pi))$$

Wie man sieht wird in dieser Definition im Gegensatz zu der vorigen Definition mit
$r = y - X(X^tV^{-1}X)^{-1}X^tV^{-1}y$ anstatt $(y - X\beta)$ gearbeitet, um die Residuen zu berechnen. Dies ist eine spezialisierte Definition, um die Struktur der Designmatrix X und der zufälligen Effekte $WV^{-1}W^t$ zu berücksichtigen.
$l_{REML}(Q, R)$ ist die eingeschränkte log-Likelihood, der sich aus der Methode 'Restricted Maximum Likelihood' ergibt und entspricht im Wesentlichen der normalen log-Likelihood mit Ausnahme einer Differenz. Bei der 'Restricted Maximum Likelihood' werden im Gegensatz zu der Methode 'Maximum Likelihood' die Freiheitsgrade, die für die Schätzung fester Effekte bei der Schätzung von Varianzkomponenten verwendet werden, berücksichtigt. Im Gegensatz zum ursprünglichen Datenvektor basiert die eingeschränkte Maximum-Likelihood-Methode auf linearen Kombinationen der Beobachtungen, die so gewählt sind, dass diese Kombinationen invariant zu den Werten der festen Effektparametern sind.
Diese linearen Kombinationen sind äquivalent zu den Residuen, die nach der Anpassung durch normale kleinste Quadrate (gewichtet bei Angabe einer Regressionsgewichtung) lediglich den festen Effektanteil des Modells berechnen. Das Verfahren führt somit eine Maximierung in einem eingeschränkten Vektorraum durch.

2.3.2 Likelihood-Ratio-Test

Die Berechnung der Likelihood-Ratio-Test-Statistik (LRT-Statistik) ist relativ einfach, sofern die Theorie der Likelihood Inferenz vergegenwärtigt wird. Zur Erinnerung: Der Vergleich eines reduziertes Modells mit dem vollständigen Modell dient der Evaluierung des Einflusses einer Störgröße und somit der Bewertung einer möglichen Verzerrung. Zur Durchführung dieser Analyse dient der Likelihood-Ratio-Test. Er ermöglicht den Vergleich eines einfacheren Modells (Nullmodell) mit einem komplexeren Modell (alternatives Modell), indem er die Likelihoods, bzw. die log-Likelihoods, der beiden Modelle vergleicht. Dies ist zum Beispiel nützlich um den Einfluss eines zusätzlichen Parameters zu beurteilen.
Der Likelihood-Ratio Test wird wie folgt definiert:

Definition 2.3.5 (Likelihood-Ratio-Test (LRT))**.**
Sei $\mathbf{L}_{full}$ der Likelihood-Wert des vollständigen Modells sowie $\mathbf{L}_{red}$ der Likelihood-Wert des reduzierten Modells. Es sei i die Anzahl der Freiheitsgrade.
Dann ist die LRT Statistik gegeben durch

$$LRT(\mathbf{L}_{full}, \mathbf{L}_{red}) = 2(\log \mathbf{L}_{full} - \log \mathbf{L}_{red})$$

Sofern die Größen $\mathbf{L}_{full}$ und $\mathbf{L}_{red}$ gemäß der Definition initialisiert sind, gilt $\mathbf{L}_{full} > \mathbf{L}_{red}$. Insbesondere gilt $\log(\mathbf{L}_{full}) > \log(\mathbf{L}_{red})$. Sofern die Log-Likelihood-Werte der Modelle bereits als l_{full} und l_{red} gegeben sind, lässt sich die LRT-Statistik durch $2(l_{full} - l_{red})$ berechnen.

Für die spätere Analyse werden noch ein paar Kenngrößen wichtig sein, welche im Folgenden definiert werden.

Die LRT-Statistik folgt asymptotisch einer χ^2-Verteilung mit k Freiheitsgraden. Dabei ist k auch die Differenz der betrachteten Effekte zwischen dem Nullmodell und dem alternativen Modell.
Eine χ^2-Verteilung ist folgendermaßen definiert:

Definition 2.3.6 (χ^2-Verteilung)**.**
Sei $X_1, X_2, \ldots, X_k$ eine Folge von unabhängigen standardnormalverteilten Zufallsvariablen, also $X_i \sim N(0,1)$ für $i = 1, \ldots, k$. Dann ist die Zufallsvariable

$$Y = \sum_{i=1}^{k} X_i^2$$

χ^2-verteilt mit k Freiheitsgraden. Wir schreiben:

$$Y \sim \chi^2(k)$$

Die Wahrscheinlichkeitsdichtefunktion der χ^2-Verteilung mit k Freiheitsgraden ist gegeben durch:

$$f(x;k) = \begin{cases} \frac{1}{2^{k/2}\Gamma(k/2)} x^{k/2-1} e^{-x/2} & x > 0, \\ 0 & x \leq 0, \end{cases}$$

wobei $\Gamma(\cdot)$ die Gamma-Funktion ist.

Die χ^2-Verteilungen sind in Abbildung 2.4 veranschaulicht.

Eine weitere wihtige Komponente für die Analyse wird der Mean-Squarred-Error (MSE/Mittlere quadratische Fehler) sein, welcher die Differenz zwischen den geschätzten Werten und den tatsächlichen Werten misst. Er wird verwendet um die Qualität eines Schätzers oder eines Vorhersagemodells zu bewerten und wird wie folgt definiert (vgl. **MSE**):

Definition 2.3.7 (Mean-Squarred-Error (MSE))**.**
Seien $\hat{y}_i$,für $i = 0, ..., n$ die geschätzten Werte und y_i die tatsächlichen Werte. Dann ist der mittlere quadratische Fehler folgendermaßen definiert:

$$MSE = \frac{1}{n} \sum_{i=0}^{n} (\hat{y}_i - y_i)^2$$

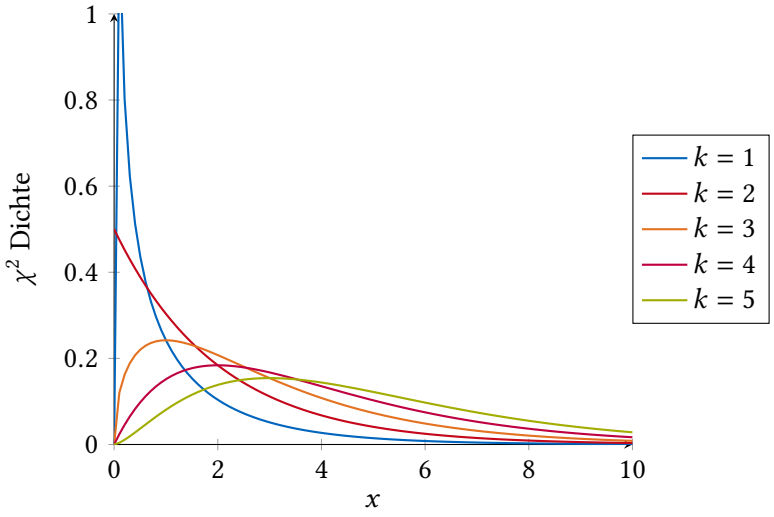


Abbildung 2.4 Chi-Quadrat-Verteilung für verschiedene Freiheitsgrade k.

3 Empirische Ergebnisse

Die nötigen theoretischen Grundlagen wurden nun für die empirischen Ergebnisse dieser Arbeit eingeführt. Im folgenden kann nun empirisch untersucht werden, ob es zu einer Verzerrung der Inferenz bei Verwendung gemischter Modelle in einer latenten Repräsentation in Form eines Variational Autoencoder kommt. Die Kombination von VAE und gemischten Modellen ermöglicht es Daten mit einer komplexen Struktur bei denen sowohl die Variabilität zwischen Gruppen als auch innerhalb von Gruppen berücksichtigt werden muss, effektiver zu analysieren. Dazu muss allerdings zuerst untersucht werden, ob es zu einer signifikanten Verzerrung kommt und gemischte Modelle im positiven Fall problemlos in latenten Repräsentationen angewendet werden können.

In diesem Kapitel wird demnach untersucht, ob es zu einer Verzerrung der Inferenz kommt. Insbesondere wird im Falle einer Verzerrung quantifiziert wie stark diese ist und ob sie eventuell akzeptiert werden kann.

3.1 Das experimentelle Modell

Für die Analyse muss zuerst das eigene Modell vorgestellt werden. In dem ersten Kapitel wurden bereits Variational Autoencoder und gemischte Modell eingeführt. In diesem Abschnitt wird nun ein gemischtes Modell auf der Encoder-Ausgabe eingeführt, um die Verzerrung der Inferenz bei solchen Anwendungen bewerten zu können. Das Modell trainiert zuerst einen Variational-Autoencoder auf einem hochdimensionalen Datensatz und daraufhin die gemischten Modelle auf der dimensionsreduzierten latenten Datenwolke des Encoder-Modells. Dazu wird nun zuerst das VAE-Modell eingeführt welches im Rahmen dieser Arbeit verwendet wurde. Seien also $q_\phi(z|x)$ die approximierte posterior Verteilung des VAE, $p_\theta(x|z)$ die Likelihood des Decoders und $p(z)$ die prior Verteilung im latenten Raum, wie sie im Kapitel 'Einführung in Variational Autoencoder' eingeführt wurden. Das Encoder Modell wird zunächst mit einer latenten Dimension und einer einzelnen verborgenen Schicht mit 150 Neuronen initialisiert. Dieser liefert den Erwartungswert μ und die Standardabweichung σ der Verteilung der Eingabedaten. Ebenso wie der Encoder wird der Decoder zunächst mit einer versteckten Schicht mit 150 Neuronen initialisiert. Beiden neuronalen Netzwerken können zusätzliche Schichten hinzugefügt werden und somit die Komplexität der Modelle erhöht werden. Der Reparametrisierungstrick in diesem VAE-Modell ist wie folgt definiert:

$$z = \mu + \exp(\log(\sigma)) \odot \epsilon$$

Dabei ist $\epsilon \sim \mathbf{N}(0, I)$. Aus ihm werden letztendlich die latenten Daten berechnet, auf denen das gemischte Modell angewendet wird.
Ein gemischtes Modell auf einer latenten Datenwolke kann, aus der Definition 2.2.3 abgeleitet, wie folgt definiert werden. Wir ersetzten dabei den y Vektor mit dem latenten Vektor z:

$$z = X\beta + Wb + \epsilon$$

mit

$$\begin{pmatrix} b \\ \epsilon \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} Q & 0 \\ 0 & R = \sigma_\epsilon^2 I \end{pmatrix}\right)$$

Das gemischte Modell wird dann durch die Maximierung der negative Log-Likelihood optimiert. Die Log-Likelihood des gemischten Modells auf der latenten Datenwolke ist dann, abgeleitet aus Definition 2.3.4, wie folgt gegeben:

$$l_{ML}(Q, R) = -0.5 * (log(|V|) + (z - X\beta)'V^{-1}(z - X\beta) + N * log(2\pi))$$

mit $V = R + WQW^t$.
Aus ihr werden dann auch die Werte für das vollständige und das reduzierte Modell für den Likelihood-Ratio-Test berechnet.

Das VAE-Modell wird nach der vorangegangenen Theorie optimiert.
Für die Berechnung der KL-Divergenz im Kontext des VAE-Modells werden die beiden Verteilungen $q_\phi(z|x)$ und $p_\theta(z|x)$ als mehrdimensionale Normalverteilungen angenommen.
Die Wahrscheinlichkeitsdichte für eine Normalverteilung wurde bereits in Definition 2.1.1 definiert. Setzt man nun in die KL-Divergenz zwischen $q_\phi(z|x)$ und $p_\theta(z|x)$, wie sie in Gleichung 2.6 definiert ist, die Wahrscheinlichkeitsdichtefunktion ein, so erhält man folgende Gleichung:

$$\mathbb{E}_{q_\phi(z|x)}\left[\log\left[\frac{q_\phi(z|x)}{p_\theta(z|x)}\right]\right] = \mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{1}{\sigma\sqrt{2\mu}}\right) - \frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 - \log\left(\frac{1}{\sqrt{2\pi}}\right) + \frac{1}{2}(x)^2\right] \tag{3.1}$$

$$= \mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{1}{\sigma\sqrt{2\mu}}\right) - \log\left(\frac{1}{\sqrt{2\pi}}\right)\right] + \mathbb{E}_{q_\phi(z|x)}\left[\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] + \mathbb{E}_{q_\phi(z|x)}\left[\frac{1}{2}(x)^2\right] \tag{3.2}$$

Der Term lässt sich durch einfache mathematische Umformungen weiter vereinfachen. Für eine genauere Herleitung siehe **KL-Div**.

$$\mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{1}{\sigma\sqrt{2\mu}}\right) - \log\left(\frac{1}{\sqrt{2\pi}}\right)\right] + \mathbb{E}_{q_\phi(z|x)}\left[\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] + \mathbb{E}_{q_\phi(z|x)}\left[\frac{1}{2}(x)^2\right] \tag{3.3}$$

$$= (-\frac{1}{2}\log(\sigma^2)) + (-\frac{1}{2}) + (\frac{1}{2}\sigma^2 + \mu^2) \tag{3.4}$$

$$= (\frac{1}{2})[-\log(\sigma^2) - 1 + \sigma^2 + \mu^2] \tag{3.5}$$

$$= (\frac{1}{2})[-2 * \log(\sigma) - 1 + \exp(2 * \log(\sigma)) + \mu^2] \tag{3.6}$$

Der Rekonstruktionsverlust des Variational Autoencoder wird durch die Differenz zwischen den Rekonstruierten Daten $\hat{x}$ und den Eingabedaten x berechnet. Dieser wird durch den Decoder ausgegeben und zusammen mit der KL-Divergenz in der Loss-Funktion berücksichtigt.

Die Loss-Funktion für das grundlegende VAE-Modell im Rahmen dieser Arbeit wird wie folgt berechnet:

$$\mathcal{L}(\alpha, \gamma) = \alpha * D_{KL}(q_\phi(z|x)||p_\theta(z|x)) + \gamma * RECLOSS$$

Dabei sind α und γ standardgemäß auf den Wert 1 ($\alpha = 1, \gamma = 1$) gesetzt. Somit entspricht die Loss-Funktion im Standardfall der negativen ELBO.

$$\mathcal{L}(\alpha = 1, \gamma = 1) = -\mathcal{L}(\theta, \phi; x)$$

Die Loss-Funktion lässt sich allerdings auch erweitern um den Einfluss des gemischten Modells zu erhöhen. Dabei wird der Mean-Squarred-Error berechnet und in der Loss-Funktion mit $\eta = 10$ gewichtet. Ebenso kann die negative Log-Likelihood der Loss-Funtkion angehängt werden. Sie wird mit $\lambda = 10$ gewichtet, sodass sich die Loss-Funktion folgendermaßen ergänzt:

$$\mathcal{L}(\alpha, \gamma, \eta, \lambda) = \alpha * D_{KL}(q_\phi(z|x)||p_\theta(z|x)) + \gamma * RECLOSS + \eta * MSE + \lambda * l_{ML}(Q, R)$$

Mit diesen Ergänzungen in der Loss-Funktion wird ein sogenanntes 'Overfitting' des Variational Autoencoder provoziert.

3.2 Experimente und Ergebnisse

Die theoretischen Grundlagen für die Ergebnisse dieser Arbeit sind nun gegeben und es kann mit der Analyse fortgefahren werden. Um am Ende die Verzerrung der Inferenz zu messen, wird im Rahmen dieser Arbeit mithilfe des Likelihood-Ratio-Tests ein vollständiges Modell mit einem reduzierten Modell verglichen. Die so erhaltene LRT-Statistik wird in einem Histogramm ausgewertet und mit einer χ^2-Verteilung abgeglichen. Die χ^2-Verteilung bietet einen Vergleichswert für die Interpretation der Ergebnisse. Somit kann festgestellt werden, wie signifikant der Einfluss des zusätzlichen Parameters ist und ob die Anwendung gemischter Modelle im latenten Raum die Inferenz verzerrt. Um einen Vergleichswert zu haben, schaffen wir zuerst ein Szenario, in dem eine χ^2-Verteilung in der LRT-Statistik erwartbar ist. Für dieses Szenario wird die Teststatistik nicht auf der latenten Repräsentation sondern auf den wahren Daten durchgeführt. Später fährt die Analyse auf einem komplexen medizinischen Datensatz in einer latenten Repräsentation fort.

3.2.1 gemischte Modelle auf simulierten Daten

Da im Rahmen der späteren Analyse ein komplexer longitudinaler medizinischer Datensatz verwendet wird, fällt die Wahl für das einfachere Szenario auf ein Simulationsdesign für einen einfachen longitudinalen medizinischen Datensatz, welchem wir dann eine Variable hinzufügen, die keinen Einfluss auf die Testergebnisse haben soll. Im Folgenden wird ein Simulationsdesign für eine Studie präsentiert, welche die Herzgesundheit von Patienten über einen Zeitraum von zehn Jahren analysiert. Die Gewichtung der verschiedenen Parameter auf den sogenannten „Health-Score" ist unterschiedlich.

Simulationsdesign
Im Rahmen einer zehnjährigen Studie wurden 500 Patienten im Alter zwischen 30 und 60 Jahren auf verschiedene Parameter untersucht, die einen Einfluss auf die Herzgesundheit haben. Die Simulationen für jeden Parameter basieren auf einer Normalverteilung und umfassen Daten über den Zeitraum von zehn Jahren. Die in Tabelle 3.1 dargestellten Einflussfaktoren sind als feste Parameter für die Herzgesundheit zu betrachten. In der Berechnung des Health-Scores wird insbesondere berücksichtigt, dass es zu zufälligen Einflussfaktoren kommen kann, die die Herzgesundheit betreffen. Daher wurde in die Berechnung ein zufälliger Interzept und eine zufällige Steigung integriert, für die eine Normalverteilung angenommen wird.

$$random\_intercept \sim \mathcal{N}(0, 2), random\_slope \sim \mathcal{N}(0, 0.1)$$

Zu Beginn der Studie wird jedem Patienten zufällig ein Alter zugewiesen, wobei die Parameter gemäß Tabelle 3.1 berechnet werden. Insbesondere wird zu einem Zeitpunkt, welcher zufällig zwischen drei und zehn Jahren für jeden Patienten festgelegt wird, die Gewichtung der Parameter angepasst. Dies soll einen Behandlungsstart mit Medikamenten simulieren. So werden dann letztendlich mit einer Health-Score Formel

$$y = 150 + gewichte * feste\_Effekte + random\_slope * jahr + random\_intercept + \epsilon$$

die Testergebnisse nach einem gemischten Modell berechnet. Dabei ist $\epsilon \sim \mathcal{N}(0, 0.1)$ ein zufälliger Fehlervektor, welcher Messfehler berücksichtigt. Eine detailliertere Beschreibung des Simulationsdesign ist im Anhang zu finden und eine beispielhafte Simulation der Daten ist in Abbildung 6.1 für 20 ausgewählte Patienten dargestellt.

LRT-Statistik
Um einen Likelihood-Ratio-Test durchzuführen, der eine Vergleichsstatistik für die spätere Analyse liefert, wird jedem Patienten zufällig ein Geschlecht zugewiesen. Das Geschlecht sollte keinen Einfluss auf die Testergebnisse haben und wird deswegen in der Berechnung des Health-Scores mit Null gewichtet.
Es wurde nun ein Szenario geschaffen, in dem ein vollständiges Modell mit einem reduzierten Modell verglichen werden kann. Das vollständige Modell berücksichtigt dabei alle festen Effekte aus Tabelle 3.1

Feste Effekte	Mittelwert	Standardabweichung	Gewicht
Systolischer Blutdruck	120	10	-0.1
Diastolischer Blutdruck	80	10	-0.1
Cholesterin	200	30	-0.2
Triglyceride	150	20	-0.2
Kreatinin	1	0.2	-0.1
Body-Mass-Index (BMI)	25	4	-0.4
Alter			-0.1

Tabelle 3.1 Einfluss und Erstellung der Parameter des Health-Scores

zuzüglich des Geschlechts, wohingegen das reduzierte Modell nur die festen Effekte aus Tabelle 3.1 berücksichtigt. Für die Erstellung der LRT-Statistik werden beide Modelle jeweils 500 Mal auf einem neu simulierten Datensatz trainiert. In jeder Iteration werden die berechneten Log-Likelihood-Werte anhand des Likelihood-Ratio-Test ausgewertet, wobei das Ergebnis der Vergleichsanalyse in einem Histogramm zusammengetragen wird. Das Histogramm wird zum Abgleich unter eine χ^2-Verteilung gelegt, welche in Abbildung 3.1a dargestellt ist.

Wie man sieht folgt das grüne Histogramm ohne Verzerrung der roten Kurve, welche die χ^2-Verteilung beschreibt. Das bedeutet, dass bis auf einzelne Ausnahmen, welche durch Instabilitäten der Berechnung immer verursacht werden können, die Ergebnisse des LRT-Statistik, wie zu erwarten, einer χ^2-Verteilung mit einem Freiheitsgrad folgen.

Data: num_simulations = 1000, X_{full}, W_{full}, X_{red}, W_{red}, y
Result: lrt_results
for $i \leftarrow 1$ ***to*** *num_simulations* **do**
  simulate the dataset according to the simulation design;
  Initialize $\delta_{full}, \delta_{red}$;
  minimize $l_{ML}(\delta_{full}) \leftarrow X_{full}, W_{full}, z$;
  minimize $l_{ML}(\delta_{red}) \leftarrow X_{red}, W_{red}, z$;
  $res_{\text{full}} \leftarrow l_{\text{ML}}(\delta_{\text{full}})$;
  $res_{\text{red}} \leftarrow l_{\text{ML}}(\delta_{\text{red}})$;
  $lrt\_val \leftarrow$ **likelihood_ratio**$(res_{\text{full}}, res_{\text{red}})$;
  $lrt\_results$ add lrt_val;
end

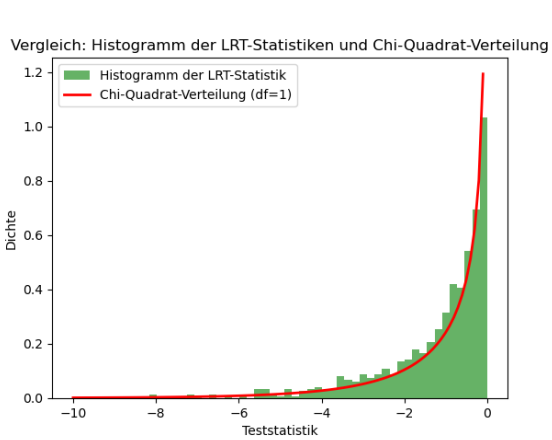
Algorithmus 1: Algorithmus zur Berechnung der Likelihood-Ratio-Teststatistik für die ge
Modelle nach dem Simulationsdesign

3.2.2 Gemischte Modelle in latenten Repräsentationen

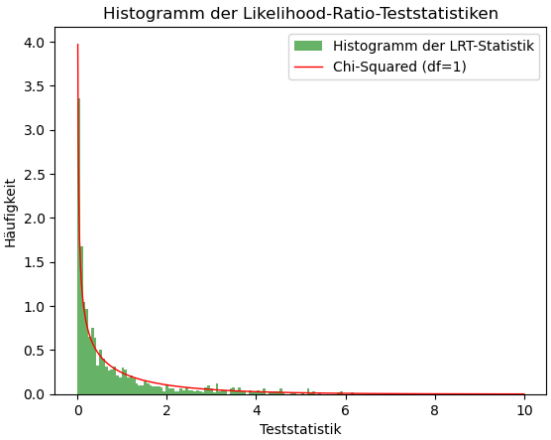
In der bisherigen Betrachtung wurden die gemischten Modelle lediglich auf Basis der realen Daten evaluiert. Im Folgenden wird die Betrachtung der gemischten Modelle auf latenten Daten vorgenommen. Zur Analyse der gemischten Modelle auf latenten Daten wird ein Variational-Autoencoder verwendet. Die Grundlage unserer Analyse bildet das folgende Grundszenario:
Zur Analyse wird nun ein komplexer longitudinaler medizinischer Datensatz betrachtet, der durch einen Encoder des VAEs im latenten Raum modelliert wird. Das Ziel ist es nun, das gemischte Modell auf dieser latenten Datenwolke zu trainieren, um herauszufinden, ob es zu einer erwartbaren Verzerrung kommt. Dazu wird im Folgenden zuerst eine detaillierte Betrachtung des vorliegenden Datensatzes vorgenommen.

Der Datensatz

Im Rahmen dieser Bachelorarbeit basieren die Ergebnisse und Experimente, um die Verzerrung der Inferenz bei der Anwendung gemischter Modelle in latenten Repräsentationen zu untersuchen, auf einem generierten, hoch-dimensionalen, medizinischem Datensatz, welcher sich aus drei zentralen Datensätzen



(a) Histogramm der LRT-Statistik mit 1000 Iterationen für gemischte Modelle auf simulierten Daten



(b) Histogramm der LRT-Statistik mit 1000 Iterationen für ein gemischtes Modell auf latenter Datenwolke

Abbildung 3.1 Histogramme der LRT-Statistiken

zusammensetzt. Diese Datensätze enthalten Informationen über die Basisdaten der Patienten, die Testergebnisse und zeitbezogene Informationen zu jedem Patienten. Zusammen ergeben sie einen komplexen Datensatz, welcher für 260 Patienten die Ergebnisse von 33 Mobilität-Tests enthält, die mehrmals wiederholt wurden. Somit lässt sich aus diesem Datensatz der Verlauf und die Schwere der Krankheit für jeden Patienten ablesen. Der Datensatz wurde aus datenschutzrechtlichen Gründen einem originellen Datensatz nachgebaut und bildet die Grundlage der Analyse dieser Arbeit.

Im Folgenden wird der komplexe medizinische Datensatz, der für die Hauptanalyse dieser Arbeit verwendet wurde, genauer beschrieben. Er setzt sich zusammen aus drei verschiedenen Datensätzen, welche die Basisdaten, die Testergebnisse und die zeitbezogenen Daten enthalten, zusammen.

Basisdaten
Der 'baseline_df' Datensatz enthält die grundlegenden Informationen der Patienten, welche mit einer eindeutigen Patienten-ID identifiziert werden. Zu jeder Patienten-ID sind folgenden Informationen gegeben:

1. 'family_affected': Gibt an, ob die Familie vorerkrankt ist.
2. 'sco_surg': Chirurgischer Score.
3. '≤ 3': binäres Merkmal.
4. 'onset_age': Alter bei Eintritt der Krankheit.
5. 'presym_diag': Prä-symptomatische Diagnose (1: Ja, 0: Nein).
6. 'presymptomatic': Prä-symptomatischer Zustand (1: Ja, 0: Nein).
7. 'stand_lost': Gibt an, ob Patient Stehfähigkeit verloren hat (1: Ja, 0: Nein).
8. 'stand_gained': Gibt an, ob Patient Stehfähigkeit gewonnen hat (1: Ja, 0: Nein).
9. 'stand_never': Gibt an, ob Patient jemals stehen konnte (1: Ja, 0: Nein).
10. 'patient_id': Eindeutige Patienten-ID.

Eine beispielhafter Eintrag im Datensatz ist in Tabelle 3.2 wiedergegeben.

Testergebnisse

patient_id	sco_surg	≤3	onset_age	presym_diag	presymptomatic	stand_lost	stand_gained	stand_never	family_affected
0	0.0	1.0	0.039397	1.0	0.0	0.0	0.0	1.0	1.0
1	0.0	0.0	2.787249	0.0	0.0	0.0	1.0	0.0	-1.0
2	1.0	1.0	1.471984	0.0	0.0	0.0	0.0	1.0	0.0
3	0.0	1.0	1.092828	0.0	0.0	0.0	0.0	1.0	-1.0
4	0.0	0.0	13.150771	0.0	0.0	0.0	1.0	0.0	-1.0

Tabelle 3.2 Basisdaten der Patienten für Patient 0 bis 4 (baseline_df)

Der Datensatz 'test_scores' enthält die Ergebnisse von insgesamt 33 Tests, in denen die Patienten einen Score zwischen 1 und 6 erreichen können. Die Spalte eines Patienten besitzt einen Mobilitäts-Wert und zu jedem Test einen Eintrag (vgl. 3.3).

patient_id	mobility	test1	test2	test3	test4	test5	test6	test7	test8	...
0	3	2	2	2	2	2	2	2	2	...
0	6	2	2	2	2	2	2	2	2	...
0	6	2	2	2	2	2	2	2	2	...
0	6	2	2	2	2	2	2	2	2	...
0	6	2	2	2	2	2	2	2	2	...

Tabelle 3.3 Testergebnisse von Patient 0 (test_scores)

Zeitbezogene Daten

Der letzte Datensatz 'time_df' enthält zeitbezogene Informationen, wie das Alter. Des Weiteren gibt er an, seit wann ein Patient behandelt wird ('since_medication') und wieviel Zeit nach dem letzten Medikamentenwechsel vergangen ist ('since_switch'). Symbolisch für den Datensatz werden die Daten für die ersten zwei Patienten in Tabelle 3.4 erfasst.

patient_id	since_medication	since_switch	age
0	1.467488	0.000000	4.346177
0	1.793292	0.000000	4.671981
0	2.447639	0.000000	5.326328
0	2.773443	0.000000	5.652132
0	3.214237	0.383299	6.092926

Tabelle 3.4 Zeitbezogene Daten von Patient 0 (time_df)

Die Designmatrizen der festen Effekte werden aus dem Basisdatensatz gewonnen, während die Designmatrizen der zufälligen Effekte aus den zeitbezogenen Daten gewonnen werden. Der Datensatz mit den Versuchsergebnissen ist das Rekonstruktionsobjekt, das dem Encoder als Eingangsdaten gegeben wird und das der Decoder so genau wie möglich zu rekonstruieren versucht. Eine genauere Beschreibung der Datensätze findet sich im Anhang.

Gemischtes Modell auf der latenter Datenwolke

Nachdem der Datensatz, mit dem im Rahmen dieser Arbeit gearbeitet wird, eingeführt wurde, kann die eigentliche Analyse dieser Arbeit beginnen. Das Ziel ist es herauszufinden, ob und wie stark es zu einer Verzerrung unter Anwendung gemischter Modelle in latenten Repräsentationen kommt. Dass die LRT-Statistik zwischen einem Nullmodell (vollständiges gemischtes Modell) und einem alternativen reduzierten gemischten Modell auf einem simulierten Datensatz einer χ^2-Verteilung folgt wurde bereits gezeigt.

Nun sollen beide gemischten Modelle auf einer latenten Datenwolke trainiert werden. Wie bereits erwähnt wird im Rahmen dieser Arbeit mit einem in Kapitel 1 'Das eigene Modell' eingeführten Variational Autoencoder (VAE) gearbeitet. Das gemischte Modell, mit welchem in dieser Arbeit gearbeitet wird, wurde bereits ebenso in Kapitel 1 eingeführt. Allerdings sind die Designmatrizen abhängig von dem Datensatz, mit dem gearbeitet wird. Deswegen werden die gemischten Modelle nochmal bezüglich des oben beschriebenen Datensatz spezifiziert.

Das experimentelle Modell
Für die Berechnung der Schätzer benötigt es natürlich die Designmatrizen für die festen und die zufälligen Effekte.
Die festen Effekte des vorgestellten Datensatzes sind alle in Tabelle 3.2 aufgezählten Parameter und der künstlich hinzugefügte Parameter 'Geschlecht'. Aus den Werten der festen Effekte setzt sich die Designmatrix der festen Effekte für das vollständige gemischte Modell X_{full} zusammen. Dementsprechend setzt sich die Designmatrix für die festen Effekte des reduzierten Modells X_{red} nur aus den Effekten aus Tabelle 3.2. Die Designmatrizen für die zufälligen Effekte W_{full}, W_{red} beider Modelle setzten sich aus den Werten der zufälligen Effekte 'since_medication', 'since_switch' und 'intercept' zusammen.
Der Antwortvektor ist in diesem Szenario nun nicht durch den aus dem gemischten Modell berechneten Antwortvektor gegeben, sondern, wie in 3.1 beschrieben, durch die aus dem Encoder gewonnene latente Datenwolke.

Für das erste Szenario, was im Rahmen dieser Arbeit untersucht wird, gibt es mehrere Trainingsschritte, die in Algorithmus 2 dargestellt sind. In diesem Szenario wird ein gemischtes Modell auf dem latenten Output eines Variational Autoencoder Im ersten Schritt des Trainingsalgorithmus wird der VAE mit der train_vae Funktion, welche die normale Loss-Funktion $\mathcal{L}(\alpha = 1, \gamma = 1)$ maximiert, trainiert. Für das Training des Variational Autoencoder wird eine Minibatch-Training verwendet. Dies ist eine Methode die verwendet wird um generative Modelle effizient auf großen Datensätzen zu trainieren, indem der Datensatz in kleine handhabbare Teilmengen, sogenannte Minibatches, aufgeteilt wird. Eine genauere Erklärung zum Minibatch-Training ist im Anhang zu finden. Im Anschluss wird die negative Log-Likelihood des vollständigen gemischten Modells mit dem latenten Vektor z und den vollständigen Designmatrizen X_{full} und W_{full} maximiert. Für das gemischte Modell wird der zweite Ordnung Optimierer LBFGS gewählt, da dieser effizienter bei großen Parameterräumen ist und die Berechnung somit schneller läuft. Sowohl für das vollständige als auch für das reduzierte gemischte Modell wurde der Optimierer wie folgt initialisiert:

1. lr = 0.01 (Lernrate)
2. max_iter = 200 (maximale Anzahl an Iterationen pro Optimierungsschritt)
3. max_eval = 500 (maximale Anzahl von Funktionsauswertungen pro Optimierungsschritt)
4. tolerance_grad = 1e-09 (Abbruchtoleranz bei Optimalität erster Ordnung)
5. tolerance_change = 1e-11 (Abbruchtoleranz bei Änderung von Funktionswerten/Parametern)
6. history_size = 200 (Größe der Update-Historie)

Mit dieser Initialisierung werden numerische Fehler während der Berechnung verhindert. Für den Variational Autoencoder fällt die Wahl auf den für tiefe neuronale Netzwerke effizienten ADAM-Optimierer, bei dem nur die Lernrate auf 0.01 angepasst wird. Ansonsten wird mit der Standardinitialisierung gearbeitet. Im zweiten Trainingsschritt wird nun der VAE mit der Funktion train_vae_2, welche den Mean Squarred Error und die negativen Log-Likelihood in der Loss-Funktion $\mathcal{L}(\alpha = 1, \gamma = 1, \eta = 10, \lambda = 10)$ hat, trainiert. Das bedeutet der Einfluss des Mean-Squarred-Error und der negativen Log-Likelihood ist sehr hoch, was das VAE-Modell provozieren könnte, zu gut zu lernen und somit in der Inferenz eine Verzerrung verursachen. Im Anschluss auf das zweite Training des Variational Autoencoder wird wieder das vollständige gemischte Modell wie zuvor trainiert. Der zweite Trainingsschritt wird dann insgesamt für 30 Iterationen

wiederholt. Sobald dieser zweite Trainingsschritt abgeschlossen ist, wird dann das reduzierte Modell noch ein Mal auf dem zuletzt berechneten latenten Vektor wie zuvor das vollständige gemischte Modell trainiert.
Die zuletzt berechnete negative Log-Likelihood des vollständigen Modell $\mathbf{L}_{full}$ und der negative Log-Likelihood Wert des reduzierten Modell $\mathbf{L}_{red}$ werden letztendlich mit dem
Likelihood-Ratio Test $LRT(\mathbf{L}_{full}, \mathbf{L}_{red})$ ausgewertet. Dieses Szenario wurde für 1000 Wiederholungen durchgeführt und die Ergebnisse in einem Histogramm in Abbildung 6.3 gesammelt.

Wie man in Abbildung 6.3 sieht, wurde das Histogramm unter eine χ^2-Verteilung gelegt und verglichen. Es ist leicht zur erkennen, dass die Teststatistik der Verteilung folgt.

Für dieses Testszenario ist demnach keine Verzerrung in der Inferenz zu erkennen, was bedeutet, dass der Mean Squarred Error und die negative Log-Likelihood trotz hoher Gewichtung in der Loss-Funktion keinen Einfluss auf das Training des Variational Autoencoder haben und der Encoder nicht zu gut lernt.

Für ein zweites Szenario, was im Rahmen dieser Bachelorarbeit untersucht wurde, wurde das VAE-Modell ausgetauscht durch einen üblichen Autoencoder. Diese Anpassung gelingt relativ einfach, da man dazu einfach den Output des Encoders nimmt und den Reparametrisierungstrick weglässt. Die latenten Variablen sind dann einfach μ aus dem oben vorgestellten VAE-Modell. Ziel ist es das Modell zu provozieren zu gut zu lernen, weswegen die Loss-Funktion aus dem VAE-Modell dem Autoencoder übergeben wird. In Algorithmus 2 müssen demnach nur die train_vae und train_vae_2 Funtkion angepasst werden. So erhält man eine neue LRT-Statistik, welche in Abbildung () dargestellt ist.

Data: num_simulations = 1000, iterations = 30, X_{full}, W_{full}, X_{red}, W_{red}
Result: lrt_results
for $i \leftarrow 1$ ***to*** *num_simulations* **do**
  Load and prepare datasets;
  Initialize *encoder*, *decoder* ;
  Initialize $optimizer\_vae(encoder.parameters(), decoder.parameters())$;
  $z \leftarrow$ **train_vae**$(epochs = 2, batch\_size = 128, encoder, decoder)$;
  for $j \leftarrow 0$ ***to*** *iterations* **do**
    if $j \neq 0$ **then**
      $z \leftarrow$ **train_vae_2**$(epochs = 1, batch\_size = 128, encoder, decoder, optimizer\_vae, W_{\text{full}}, X_{\text{full}}, Q_{\text{full}}, R_{\text{full}})$;
    end
    Initialize Q_{full}, R_{full};
    Initialize $optimizer\_mm\_full(Q_{red}, R_{red})$ **maximize** $l_{ML}(Q_{full}, R_{full}) \leftarrow X_{\text{full}}, W_{\text{full}}, z$;
  end
  Initialize Q_{red}, R_{red};
  Initialize $optimizer\_mm\_full(Q_{red}, R_{red})$ **maximize** $l_{ML}(Q_{red}, R_{red}) \leftarrow X_{\text{red}}, W_{\text{red}}, z$;
  $res_{\text{full}} \leftarrow l_{\text{ML}}(Q_{\text{full}}, R_{\text{full}})$;
  $res_{\text{red}} \leftarrow l_{\text{ML}}(Q_{\text{red}}, R_{\text{red}})$;
  $lrt\_val \leftarrow$ **likelihood_ratio**$(res_{\text{full}}, res_{\text{red}})$;
  $lrt\_results$ add lrt_val;
end

Algorithmus 2: Algorithmus zur Simulation und Berechnung der Likelihood-Ratio-Teststatistik aus Experiment 1

3.2.3 Post-Selection-Inferenz (PSI)

Post-Selection Inference (PSI) bezieht sich auf statistische Methoden, die die Tatsache berücksichtigen, dass eine Hypothese oder ein Modell aufgrund der Daten ausgewählt wurde. Traditionelle statistische In-

ferenzmethoden setzen oft voraus, dass das Modell oder die Hypothese a priori festgelegt wurde, was in der Praxis selten der Fall ist. Bei PSI werden die statistischen Eigenschaften angepasst, um die zusätzliche Unsicherheit durch den Selektionsprozess zu reflektieren. Dies ist besonders wichtig in Bereichen wie maschinellem Lernen und Data Mining, wo oft zahlreiche Modelle getestet und das beste ausgewählt wird. Durch PSI können zuverlässigeren Konfidenzintervalle und p-Werte berechnet werden, die die Selektion berücksichtigen, was zu robusteren und aussagekräftigeren Ergebnissen führt. Verfahren zur Untersuchung der Verzerrung in den Inferenzergebnissen.

Einführung

Die Post-Selection-Inferenz (PSI) ist eine statistische Methode, die darauf abzielt, die Verzerrungen zu korrigieren, die nach der Modellauswahl auftreten können. Traditionelle Inferenzmethoden gehen davon aus, dass das Modell a priori festgelegt ist. In der Praxis wird jedoch häufig eine Modellauswahl basierend auf den Daten durchgeführt, was zu einer systematischen Verzerrung der Inferenz führen kann.

PSI berücksichtigt diese Modellauswahl und bietet Methoden zur Korrektur der dadurch entstehenden Verzerrungen. Insbesondere bei komplexen Modellen wie Variational Autoencoders (VAE) in Kombination mit gemischten Modellen ist die Berücksichtigung der Modellauswahl von entscheidender Bedeutung, um zuverlässige und gültige Schlussfolgerungen ziehen zu können.

Prinzipien der Post-Selection-Inferenz

PSI basiert auf der Idee, die Unsicherheit der Modellauswahl explizit in die Inferenz einzubeziehen. Dies wird erreicht durch:

Konditionierung auf die Modellauswahl: Anstatt die Inferenz auf dem ausgewählten Modell durchzuführen, wird die Unsicherheit, die durch den Auswahlprozess entsteht, konditioniert. Dies bedeutet, dass die Schätzungen und Tests die Tatsache berücksichtigen, dass das Modell aus einer Menge möglicher Modelle ausgewählt wurde. Anpassung der Teststatistiken: Die Teststatistiken und Konfidenzintervalle werden angepasst, um die zusätzlichen Freiheitsgrade, die durch die Modellauswahl entstehen, zu berücksichtigen. Dadurch werden konservativere und weniger verzerrte Schätzungen erzielt.

3.2.4 Anwendungen der Post-Selection-Inferenz

Die Anwendung von PSI in der Kombination von VAEs und gemischten Modellen kann mehrere Vorteile bieten:

Verbesserte Zuverlässigkeit der Inferenz: Durch die Berücksichtigung der Modellauswahlprozesse kann die Verzerrung der Inferenz reduziert werden, was zu verlässlicheren und stabileren Ergebnissen führt. Genaue Konfidenzintervalle: PSI bietet genauere Konfidenzintervalle für die Modellparameter, die die Unsicherheit der Modellauswahl korrekt widerspiegeln. Bessere Modellbewertung: Die Anpassung der Teststatistiken ermöglicht eine genauere Bewertung der Modellgüte und eine verbesserte Entscheidungsfindung bei der Auswahl des besten Modells.

Ausblick

Die Integration von PSI in die Analyse von gemischten Modellen in latenten Repräsentationen eröffnet neue Möglichkeiten zur Verbesserung der statistischen Inferenz. In zukünftigen Arbeiten könnte die Anwendung von PSI auf verschiedene Datensätze und Modellkonfigurationen weiter erforscht werden, um die Robustheit und Genauigkeit der Inferenz zu erhöhen. Dies könnte insbesondere in Bereichen wie der medizinischen Datenanalyse, wo präzise und zuverlässige Inferenzmethoden von entscheidender Bedeutung sind, von großem Nutzen sein.

3.2.5 Analyse der Ergebnisse

Wie man sieht folgt das grüne Histogramm ohne Verzerrung der roten Kurve, welche die χ^2-Verteilung beschreibt. Das bedeutet, dass bis auf einzelne Ausnahmen, welche durch Instabilitäten der Berechnung immer verursacht werden können, die Ergebnisse des LRT-Statistik, wie zu erwarten, einer χ^2-Verteilung mit einem Freiheitsgrad folgen.

Wie man in Abbildung 6.3 sieht, wurde das Histogramm unter eine χ^2-Verteilung gelegt und verglichen. Es ist leicht zur erkennen, dass die Teststatistik der Verteilung folgt.

Für dieses Testszenario ist demnach keine Verzerrung in der Inferenz zu erkennen, was bedeutet, dass der Mean Squarred Error und die negative Log-Likelihood trotz hoher Gewichtung in der Loss-Funktion keinen Einfluss auf das Training des Variational Autoencoder haben und der Encoder nicht zu gut lernt.

Diskussion der Ergebnisse im Hinblick auf die Verzerrung der Inferenz.

4 Diskussion und Fazit

4.1 Interpretation der Ergebnisse

Tiefere Analyse der Ergebnisse und ihrer Implikationen.

4.2 Vergleich mit bestehenden Arbeiten

Wie sich die Ergebnisse zu bereits veröffentlichten Forschungen verhalten.

4.3 Limitationen und Herausforderungen

Kritische Betrachtung der Grenzen der Studie und mögliche Probleme.

5 Fazit

Diese Arbeit hat die Verzerrung der Inferenz bei der Verwendung gemischter Modelle in latenten Repräsentationen untersucht, insbesondere unter Einsatz von Variational Autoencoders (VAE). Die Ergebnisse zeigen, dass die Kombination von VAEs und gemischten Modellen eine vielversprechende Methode zur Analyse hochdimensionaler Daten ist. Es konnte jedoch eine signifikante Verzerrung in der Inferenz festgestellt werden, wenn das gemischte Modell auf der latenten Datenwolke trainiert wurde. Diese Verzerrung führt zu fehlerhaften Schlussfolgerungen und mindert die Validität der Modelle. Dennoch bieten die durchgeführten Analysen wertvolle Erkenntnisse für die Anwendung und Weiterentwicklung dieser Modelle in der Praxis.

5.1 Ausblick

Für zukünftige Forschungen wird empfohlen, die Verzerrung der Inferenz weiter zu untersuchen und zu minimieren, insbesondere durch die Integration von Post-Selection-Inferenz (PSI). PSI-Methoden berücksichtigen die zusätzliche Unsicherheit, die durch den Selektionsprozess entsteht, und können somit robustere und aussagekräftigere Ergebnisse liefern. Dies ist besonders relevant in Bereichen wie dem maschinellen Lernen und Data Mining, wo häufig zahlreiche Modelle getestet und das beste ausgewählt wird. Durch die Anwendung von PSI können verlässlichere Konfidenzintervalle und p-Werte berechnet werden, was die Ergebnisse weiter absichert.

Zukünftige Forschungen sollten sich darauf konzentrieren, alternative Methoden zur Regularisierung der VAEs zu entwickeln und optimierte Trainingsverfahren für gemischte Modelle zu testen. Darüber hinaus wäre es sinnvoll, die Anwendbarkeit dieser Methoden auf reale medizinische Datensätze zu überprüfen, um deren praktischen Nutzen zu validieren und gegebenenfalls anzupassen. Es gibt Hinweise darauf, dass PSI noch ungenutztes Potenzial hat und möglicherweise mehr Möglichkeiten bietet, die Genauigkeit und Robustheit der Inferenz zu verbessern.

Die Integration von PSI in die Analyse gemischter Modelle in latenten Repräsentationen stellt einen vielversprechenden Ansatz dar, um die Zuverlässigkeit und Aussagekraft der Inferenzmethoden zu erhöhen. Dies könnte die Grundlage für weiterführende Studien und die Entwicklung neuer, fortschrittlicher Analysemethoden bilden

Zusammenfassung der wichtigsten Erkenntnisse Praktische Implikationen: Wie die Ergebnisse in der Praxis angewendet werden können. Empfehlungen für zukünftige Forschungen: Vorschläge für weiterführende oder ergänzende Studien.

6 Anhang

6.1 Herzgesundheits-Datensatz

Das Simulationsdesign für den Herzgesundheits-Datensatz wurde bereits in Kapitel 3.2.1 beschrieben. Es folgen Ergänzungen zur Veranschaulichung und zum tieferen Verständnis des Simulationsdesign. Die Trajektorien der festen Effekte 'Systolischer Blutdrck', 'Diastolischer Blutdruck', 'Cholesterinspiegel', Body-Maß-Index' (BMI) und der Testwerte 'Gesundheitsscore' sind in Abbildung 6.1 für fünf zufällig ausgewählte Patienten dargestellt. Der Wechsel der Farbe einer Trajektorie eines Patienten von Rot auf Grün symbolisiert den Anfang einer Behandlung. Es ist zu erkennen, dass sich mit dem Wechsel der Farben die Werte grundsätzlich verbessern. Dies hängt damit zusammen, dass mit Start der Behandlung die aus den Normalverteilungen (vgl. 3.1) gezogenen Daten pro Jahr mit den in Tabelle 6.1 dargestellten Werten verbessert werden. Vor der Behandlung werden die Daten mit geringeren Parametern angepasst, was eine minimale natürliche Verbesserung simulieren soll. Die Werte dazu sind genauso wie der Einfluss der festen Effekte auf den Gesundheitsscore in Tabelle 6.1 dargestellt. Der Start der Behandlung wird zufällig nach zwischen Jahr drei bis zehn ausgewählt.

	natürliche Verbesserung pro Jahr	Behandlungseffekt pro Jahr	Effekt auf den Gesundheitsscore
Systolischer Blutdruck	-0.5	-2	-0.1
Diastolischer Blutdruck	-0.5	-2	-0.1
Cholesterin	-1	-5	-0.2
Triglycerides	-1	-3	-0.2
BMI	/	/	-0.4
Creatinin	-0.01	-0.08	-0.1

Tabelle 6.1 Gewichte des Simulationsdesigns des Herzgesundheits-Datensatz

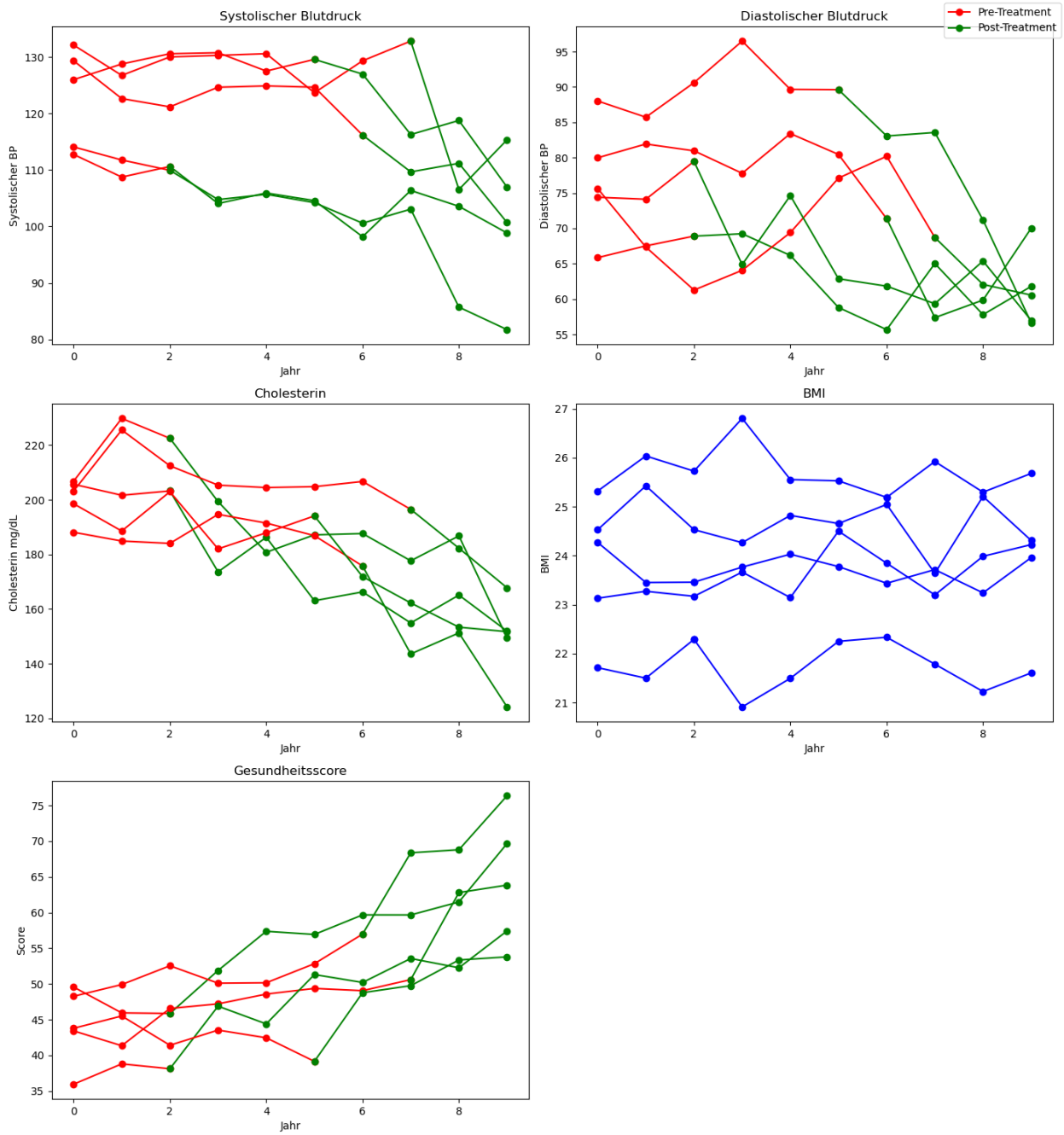


Abbildung 6.1 Trajektorien der festen Effekte und des Gesundheitsscores eines simulierten Datensatzes für 20 zufällig ausgewählte Patienten

6.2 komplexer medizinischer Datensatz

Die LRT-Statistik für das in Kapitel 3.2.2 beschriebene Modell wurde vorerst ohne die negative Log-Likelihood in der Loss-Funktion erstellt, um zuerst den Einfluss des Mean Squarred Error zu testen. Das Resultat ist in Abbildung 6.2 dargestellt. Da dieser auch bei hoher Gewichtung in der Inferenz keine Verzerrung verursacht hatte, wurde die negative Log-Likelihood hinzugefügt, um ein zu gutes Training stärker zu provozieren. Wenn man die Teststatistiken in Abbildung 6.2 und Abbildung 6.3 vergleicht sind keine größeren Unterschiede zu erkennen.

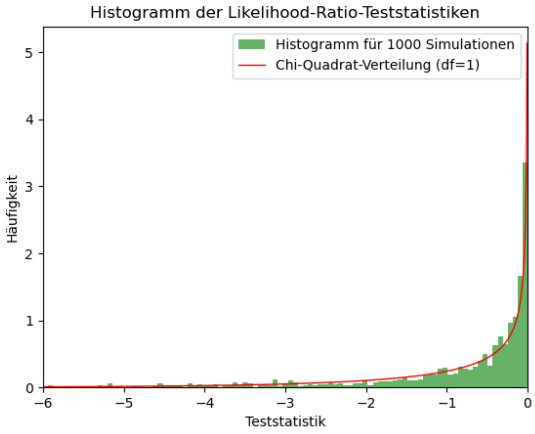


Abbildung 6.2 Histogramm der LRT-Statistik mit $\mathcal{L}(\alpha = 1, \gamma = 1, \eta = 10, \lambda = 0)$

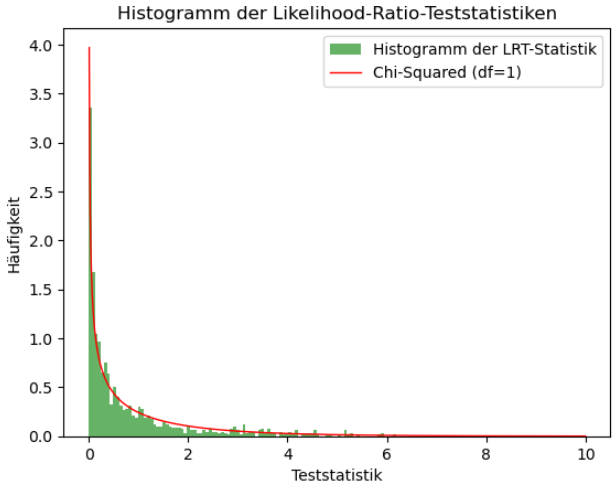


Abbildung 6.3 Histogramm der LRT-Statistik mit $\mathcal{L}(\alpha = 1, \gamma = 1, \eta = 10, \lambda = 10)$

6.3 Minibatch-Training

Das Minibatch-Training ist eine Form des Stochastic Gradient Descent (SGD), bei dem die Modellparameter mithilfe kleiner, zufällig ausgewählter Teilmengen des Datensatzes aktualisiert werden, anstatt den gesamten komplexen Datensatz auf einmal zu verwenden. Diese Methode hat mehrere Vorteile. Es reduziert nicht nur den Speicherbedarf, da immer nur kleine Teilmengen der Daten im Speicher geladen werden, sondern ermöglicht gleichzeitig eine schnellere Konvergenz, da die Modellparameter häufiger aktualisiert werden.

Im Falle des in dieser Arbeit verwendeten Modells, wird vor der Trainingsschleife der Datensatz für ein Minibatch-Training vorbereitet. Dies geschieht ganz einfach, indem der Datensatz in mehrere Minibatches aufgeteilt wird. Die Größe der Minibatches ist häufig eine Potenz von 2 (z.B. 16, 32, 64, 128)

A Appendix

A.1 Supporting Data

A.2 Some Code Listings

Danksagungen

An dieser Stelle möchte ich mich recht herzlich bei Herr Clemens Schächter für die stets zuvorkommende und zeitintensive Betreuung bedanken. Ohne den ständigen Austausch und ohne die interessanten Anregungen wäre diese Arbeit mit Sicherheit so nicht zustande gekommen.

Außerdem bedanke ich mich recht herzlich bei Prof. Dr. Harald Binder für die sehr angenehme Kooperation und den häufigen fachlichen Austausch.

Erklärung

Hiermit erkläre ich, dass ich diese Bachelorarbeit eigenständig verfasst habe und keine anderen als die von mir angegebenen Quellen und Hilfsmittel benutzt habe und dass ich diese Arbeit nicht bereits zur Erlangung eines akademischen Grades eingereicht habe.