筑波大学社会・国際学群国際総合学類

卒業論文

GenSQLを用いた合成データ生成におけるデータ量と精度評価

2025 年 1 月

氏名：岩佐克哉

学籍番号：202110584

指導教員：岡瑞起

要旨

データ数と合成データの精度には明確な関係性があることが確認された。まず、多量データでは、合成データ生成手法である GenSQL が元データの分布特性を忠実に再現できることが示された。たとえば、データ数が 150,000 や 10,000 といった大規模な条件下では、合成データと元データ間の TVD（Total Variation Distance）は 0.05〜0.1 に収まり、標準偏差も比較的低い水準であった。これは、元データの統計的特徴をモデルが十分に捉え、分布特性を安定して再現できることを示唆している。こうした条件下では、合成データは元データに非常に近い分布的整合性を持ち、代替データとして十分に活用可能である。

さらに、多量データ（データ数が 10,00 以上）および中程度のデータ（データ数が 10,000 から 1,000）においては、合成データ生成の際に設定する minutes パラメータが結果に与える影響も無視できないことがわかった。具体的には、生成時間を短く設定すると信頼区間が広がる一方で、TVD が低下する傾向が見られたが、minutes=1 のような極端に短い場合には、TVD が低くなり、信頼区間も広がってしまうため、生成データの精度と安定性に課題が生じていた。一方、minutes=60 では最も高い精度が得られ、信頼区間も狭く、安定したデータ生成が可能であることが確認された。また、minutes=10 に設定すると minutes=1 よりは TVD が大きく向上し、効率性と精度のバランスを保つ点で実用的な選択肢となる。

一方、データ数を段階的に削減した結果、少量データ（データ数が 1,000 以下）では合成精度が著しく低下することが明らかになった。特にデータ数が 1,000 未満になると、TVD は 0.1 を超え、500 や 100 といった極端に少ない条件では 0.2〜0.3 台に達した。さらに、標準偏差が増大し、試行ごとのばらつきが大きくなる傾向も確認された。この現象は、モデルがデータ数不足のため分布特性を正確に推定できず、出現頻度の低いカテゴリや特徴範囲を適切に再現するための情報が不足していることに起因していると考えられる。こうした少量データでは minutes パラメータの調整による影響が相対的に小さく、生成時間を延ばしても精度を飛躍的に向上させることは難しいことも分かった。

今後の課題としては、少量データからでも高品質な合成データを生成するためのモデル改善が挙げられる。さらに、評価指標の多様化やタスクベースの実践的評価を通じて、合成データの品質を多角的に検証する必要がある。また、フェデレーションラーニングや差分プ

ライバシーといった新しい技術との統合、異なるデータ種類や応用分野での再検証を進めることで、合成データ生成技術の汎用性と実用性をさらに向上させることが期待される。

目 次

第1章　序論

1.1　背景

近年、顧客属性データや購買履歴などの大規模個人関連データは、マーケティング、ビジネス、公共政策など多方面で有用なインサイトを生み出している。これらのデータは、精緻な顧客セグメンテーション、ターゲティング戦略、需要予測、リスク評価といった多様な目的に活用され、意思決定を支える重要な資産である。

しかし、データ活用の成否はデータの質や量に大きく依存しており、それらが十分に担保されていない場合、効果的な活用が難しい現状がある。また、プライバシー保護や社外秘情報の取り扱い規則、法規制が強化される中で、センシティブなデータの取り扱いや活用は依然として制約を受けている。その結果、元データを直接第三者に開示することが困難であり、十分な成果を上げるためには新たな手法やツールの導入が必要とされている。

こうした課題に対し、マサチューセッツ工科大学によって提案された手法の一つが、GenSQL [1] である。GenSQL はラージポピュレーションモデルを生成し、それを基に推論や合成データ生成を行うデータ解析ツールの一つである。ラージポピュレーションモデルとは、膨大なデータ群の統計的特性を捉えるモデルのことであり、医療データやソーシャルネットワークデータなどの分野において、データの自動生成、プライバシー保護、条件付き推論のために利用される。例えば、統計的に一貫性のある合成データを生成しながら個人情報を保護するための差分プライバシー手法 [2] や、統計モデルの構築と推論 [3] にも用いられている。GenSQL はテーブルデータ解析において有用なツールであるが、開発段階にあるため、データの有効活用が期待される一方で、実用化における課題も存在している。

GenSQL を用いたモデル構築や合成データ生成に関しては、データ数の変化がクエリ処理時間に与える影響を評価する実験が行われている。一方で、データ数の変化や minutes パラメータが合成データ生成の精度に与える影響を評価する実験は未実施である。[1]

1.2 研究目的

既存の研究では、CTGAN [4] などの合成データ生成手法が提案されているものの、確率的プログラミング言語を用いた GenSQL のような手法が、データ数や minutes パラメータの増減ででどの程度性能を維持できるかについての詳細な定量分析は十分に行われていなかった。本研究は、このような研究のギャップを埋めることを目的としている。

本研究では、確率的プログラミングを用いたテーブルデータ合成手法である GenSQL を対象に、その性能を分析する。まず、データ数を段階的に削減した場合、合成データの元データとの分布再現性がどのように変化するかを Total Variation Distance（TVD）による評価を通じて明らかにする。次に、minutes パラメータを増減することで、TVD がどの程度改善可能かを調査し、minutes パラメータが性能に与える影響を定量的に検証する。

1.3 本論文の構成

本論文では、第 2 章において、関連研究として確率的プログラミング、ベイズ推定、MCMC、GenSQL について概説する。GenSQL は確率的プログラミングを基盤とし、ベイズ推定や MCMC を活用して条件付き分布の推定や合成データ生成を実現しているため、これらの技術との関連性を明らかにする。続いて、第 3 章ではこれらの関連研究を踏まえ、データセットの概要、およびデータの前処理や評価指標の算出方法、minutes パラメータによる推論時間の制御、GenSQL の実験設定とパラメータについて述べる。データ数、カテゴリ数、minutes パラメータによる結果への影響について第 4 章で示す。また同章で考察と今後の展望についても言及する。第 5 章では、本研究のまとめを行う。

第2章　関連研究

2.1　確率的プログラミング

近年、複雑な確率モデルの構築と推論を容易にする「確率的プログラミング（Probabilistic Programming）」が、機械学習・統計学分野において注目を集めている [5]。Stan [6] や PyMC3 [7]、TensorFlow Probability [8] など代表的な確率的プログラミングフレームワークは、確率モデルを高水準なプログラミング言語で記述し、MCMC（Markov chain Monte Carlo）や変分推論など多彩なベイズ推論手法を自動実行可能な環境を提供している [6]。こうしたフレームワークは、推論手続きやモデル設計を明示的にプログラムとして記述できるため、モデリング工程の再利用性や拡張性を高め、ユーザが統計的背景を十分に理解していなくても複雑なモデルを容易に実験することを可能にする。

2.1.1　ベイズ推論

ベイズ推定は、ベイズの定理に基づき、観測データから未知のパラメータの事後分布を推定する手法である。ベイズの定理は以下のように表される

$$P(\theta \mid \mathbf{D}) = \frac{P(\mathbf{D} \mid \theta)P(\theta)}{P(\mathbf{D})} \tag{2.1}$$

- $P(\theta \mid \mathbf{D})$ は事後分布（観測データ $\mathbf{D}$ を基にしたパラメータ θ の確率分布）
- $P(\mathbf{D} \mid \theta)$ は尤度関数（パラメータ θ の下で観測データ $\mathbf{D}$ が得られる確率）
- $P(\theta)$ は事前分布（観測前のパラメータ θ の確率分布）
- $P(\mathbf{D})$ は規格化定数

この枠組みにより、観測データが得られる前の事前情報を統合し、観測データから得られる新しい情報を反映した推定を行うことができる。

ベイズ推定の特徴としてまず挙げられるのは、事前分布を活用できる点である。過去の知識や専門的な情報を事前分布として取り入れることにより、観測データが限られている状況でも有用な推定が可能となる。次に、ベイズ推定は不確実性の定量化を実現するという重要な特性を持つ。推定結果が確率分布として表現されるため、得られたパラメータに対する信頼性や不確実性を評価することができる [9]。また、ベイズ推定は柔軟なモデル化が可能であり、複雑なデータ構造や非線形な関係を持つモデルにも適用できる点で他の推定手法と一線を画している [10]。

近年、ベイズ推定は統計モデリングや機械学習の分野で幅広く活用されている。その中でも、マルコフ連鎖モンテカルロ法（MCMC）や変分ベイズ法といった計算手法の進展により、高次元データや複雑なモデルにおいても効率的に事後分布を推定できるようになった [11]。このような技術的進化により、ベイズ推定は因果推論や時系列解析、医療分野における診断モデルの構築など、多岐にわたる応用分野で重要な役割を果たしている [9]。

2.1.2 MCMC(Markov Chain Monte Carlo)

マルコフ連鎖モンテカルロ法（MCMC）は、統計モデルに基づくデータ生成や、確率分布の推定において広く用いられている手法である [11]。特に、複雑な事後分布のサンプリングが必要な場合において、MCMC は効率的かつ柔軟な手法として評価されている [12]。MCMC は反復的に確率分布に従うサンプルを生成することにより、高次元分布の特徴を効率的に捉えることが可能である [9]。

マルコフ連鎖は、確率過程の一種で、次の状態が現在の状態のみに依存する特性（マルコフ性）を持つ。この過程では、過去の情報は現在の状態に集約されるため、システムの振る舞いをシンプルにモデル化できる。マルコフ連鎖は状態空間内を遷移しながら動き、遷移確率行列によって次の状態への移行が定義される。

モンテカルロ法は、乱数を利用して数値的な解を近似する手法である。具体的には、目標とする分布や関数の特徴をランダムサンプリングを通じて推定する。例えば、積分や確率分布の期待値の近似に用いられる。

近年、これらの理論を基盤とした実用的な確率的プログラミングツールやライブラリが数多く登場している。代表的な例として、高度なベイズモデリングと統計推論を可能にする Stan [6]、Python ベースで簡潔かつ直感的なモデリングを提供する PyMC3 [7]、そしてディープラーニングフレームワークと統合可能な TensorFlow Probability [8] が挙げられる。

これらのツールは、統計モデルの構築やデータ解析を効率的に行うためのフレームワークを提供している。

確率的プログラミングを活用することで、データに内在する不確実性を明示的に取り扱うことが可能となる。特に、高次元のテーブルデータや欠損データ、さらにはノイズの多いデータに対して有効であり、強力な解析手法を実現できる点が特徴である。これにより、テーブルデータ解析における欠損値の補完、異常検知、クラスタリング、回帰分析など、さまざまな用途に対応可能である。さらに、複雑なデータ構造を扱う際にも、確率的プログラミングは極めて有用な技術と位置付けられている。

2.2 GenSQL

GenSQL とは、マサチューセッツ工科大学によって提案された確率的プログラミングの枠組みを活用したデータ生成技術である。GenSQL はテーブルデータに対する確率的問い合わせを SQL ライクなインターフェースで記述し、それに基づいて合成データを生成できる点で特徴的である。GenSQL では、ベイズ的な確率モデルを内部的に構築し、欠損値推定や条件付き分布サンプリングをはじめとする統計的タスクを SQL 拡張文法で記述可能にすることで、ユーザが高レベルなクエリベースのインタラクションを通じて複雑な分布構造を扱うことができる [1]。

従来、合成データ生成には、モンテカルロ法や変分推論によるモデル推定を手動で行い、その上でサンプル生成を行う一連の作業が必要だった。しかし、GenSQL は確率的プログラミング技術を背後に持つことで、これらの作業を統合的かつ自動的に実現する。また、SQL に近似した記法を用いることで、データベース操作に精通した実務者や分析者が、統計的・機械学習的専門知識を必ずしも深く持たずとも、合成データ生成タスクに参入しやすくなる点も大きな利点である。

既存の SQL の機能を拡張し、確率モデルを統合的に操作するため以下の四つの構文を加えている。

```
GENERATE UNDER m
```

これは、確率モデル m に基づき合成データを生成するためのものである。指定されたモデルの分布に従った新たなデータ行を生成し、シミュレーションや新データの作成に利用できる。

```
m GIVEN e
```

これは、確率モデル m を特定の条件またはイベント e に基づき条件付けを行うためのものである。この操作により、条件 e を満たす分布に基づいた新たなモデルが生成され、ターゲットデータ生成やシナリオ分析に適用可能である。

```
t GENERATIVE JOIN m
```

これは、データテーブル t の各行と確率モデル m から生成された合成データを結合するものである。この句の特徴は、テーブル t の各行の値を基に確率モデルからのデータ生成を条件付けできる点にある。これにより、データの補完や拡張、特定条件を満たすデータ生成が実現される。

```
PROBABILITY OF e UNDER m
```

これは、確率モデル m の下で特定のイベント e の確率または確率密度を計算するものである。構文のように記述することで、モデルに基づくイベント e の発生確率を評価することが可能となる。これは、異常検知や確率推定、ベイズ推論などのタスクにおいて重要な役割を果たす。

第3章　提案手法

本研究で提案する手法は、ユーザ顧客データを用いて合成データを生成し、その生成精度を分布的な観点から評価するものである。顧客データには性別、年齢、購入履歴、居住地域、会員ステータスなど、マーケティングや行動分析に多面的に応用可能な多様な特徴量が含まれている。しかし、顧客データはしばしばプライバシー保護やデータ共有の制約から、容易に外部と交換できない問題がある。本研究の手法では、元データの分布特性を忠実に保った合成データを自動生成可能な GenSQL を用いることで、元データ自体を渡すことなく、分析可能な代替データを提供することを目指す。また、元データ行数を 150,000 行から 100 行まで段階的に減らすことで、データ数削減に伴う合成精度の低下やその限界点を明らかにする。

3.1　データセット

本研究で使用する博報堂が提供する顧客データセットは、全体で 200,000 行のレコードを有している。このうち 50,000 行は合成データ生成やモデル学習には用いず、評価目的のテストデータとして別途保存しておく。一方、残りの 150,000 行を合成データ生成の元データとして利用し、GenSQL を用いて合成データを生成する。これにより、生成過程で利用したデータと評価用データを明確に分離し、合成手法が元データ分布をどの程度適切に再現できているかを公平かつ客観的に検証することが可能となる。

データセットには以下の項目が含まれており、ID、性別、年代別区分、居住地域、婚姻状況、子供の有無、世帯所得階層、個人所得階層、職業区分、所属産業分野、テレビメディアへの接触頻度、1 日あたりのテレビ視聴時間など、多面的な顧客属性が収集されている。なお、今回の実験ではカラム数は 12 個に絞って検証を行った。

今回の実験で使用するカラムは以下のとおりである。

1. ID：顧客を一意に識別する ID

2. SEX：性別（1：男性、2：女性）

3. AGEID：年代別区分（1：12 歳未満、2：12〜19 歳、3：20〜24 歳、4：25〜29 歳、5：30〜34 歳、6：35〜39 歳、7：40〜44 歳、8：45〜49 歳、9：50〜54 歳、10：55〜59 歳、11：60 歳以上）

4. AREA：居住地域（1：北海道、2：東北、3：関東、4：中部、5：近畿、6：中国、7：四国、8：九州）

5. MARRIED：婚姻状況（1：未婚、2：既婚）

6. CHILD：子供の有無（1：なし、2：あり）

7. HINCOME：世帯所得階層（1：200 万円未満、2：200 万円〜400 万円未満、3：400 万円〜600 万円未満、4：600 万円〜800 万円未満、5：800 万円〜1000 万円未満、6：1000 万円〜1200 万円未満、7：1200 万円〜1500 万円未満、8：1500 万円〜2000 万円未満、9：2000 万円以上、10：該当なし（空欄））

8. PINCOME：個人所得階層（1：200 万円未満、2：200 万円〜400 万円未満、3：400 万円〜600 万円未満、4：600 万円〜800 万円未満、5：800 万円〜1000 万円未満、6：1000 万円〜1200 万円未満、7：1200 万円〜1500 万円未満、8：1500 万円〜2000 万円未満、9：2000 万円以上、10：該当なし（空欄））

9. JOB：職業区分（1：公務員、2：管理職・役員、3：事務系会社員、4：技術系会社員、5：一般会社員（種別不明）、6：自営業、7：フリーランス、8：専業主婦（主夫）、9：パート・アルバイト、10：学生、11：その他、12：無職）

10. Q1（Industry）：所属・関連する産業分野（1：食品・飲料、2：繊維・衣料品、3：化学・石油、4：製薬、5：鉄鋼・金属、6：自動車・輸送、7：機械、8：精密機器・電機・電子、9：製造業一般、10：卸売・小売、11：飲食業、12：金融・証券・保険、13：運輸・倉庫、14：通信・情報サービス、15：不動産（住宅含む）、16：旅行・ホテル・レジャー、17：サービス、18：農業・漁業・林業、19：建設・土木・建築、20：商社、21：公社、22：医療・福祉、23：教育、24：行政・公共機関、25：法人・団体・農協、26：その他、27：家事・求職中、28：パート・アルバイト、29：専業主婦（主夫）、30：無職（年金受給含む）、31：該当なし（空欄））

11. Q3S1（Frequency-of-TV-contact）：テレビ接触頻度（1：週 6〜7 日、2：週 4〜5 日、3：週 2〜3 日、4：週 1 日、5：月 2〜3 日、6：月 1 日、7：月 1 日未満、8：接触なし）

12. Q4S1（Daily-TV-contact-time）: 1 日あたりのテレビ視聴時間（1 : 30 分未満、2 : 30 分〜1 時間未満、3 : 1〜2 時間未満、4 : 2〜4 時間未満、5 : 4〜6 時間未満、6 : 6〜8 時間未満、7 : 8〜12 時間未満、8 : 12 時間以上）

これらの属性情報を活用することで、顧客行動の洞察やマーケティング戦略立案への活用、顧客セグメンテーション分析など、幅広い応用が期待できる。さらに、50,000 行をテストデータとして保持することで、合成データの品質評価や元データの分布の再現性の検証を厳密に行うことが可能となる。

3.2 データの前処理

分析に先立ち、以下の前処理を行う。元データから不要列を除去し、列名の正規化とデータ型の統一を行い、その上で目的変数や説明変数として必要な列のみを抽出した。結果として、欠損値に対応可能な Int64 型の名義データ群を含む解析用データセットが整備された。これらの前処理によって、GenSQL が扱いやすく、元データの分布を反映しやすい状態を整える。

3.3 評価指標の算出

本研究では、合成データが元データの分布特性をどの程度再現できているかを評価するために、確率分布間の「距離」を測定する指標として Total Variation Distance（TVD）を用いる。TVD は、2 つの確率分布 P と Q の間で定義され、

$$\mathrm{TVD}(P, Q) = \frac{1}{2} \sum |P(x) - Q(x)| \tag{3.1}$$

によって与えられる。ここで、$P(x)$ および $Q(x)$ は、元データと合成データに対応する分布上の任意のカテゴリ x における確率を示す。TVD が 0 に近い場合、元データと合成データの分布はきわめて類似していることになり、合成モデルが元データ分布を忠実に再現していることを示す。

本研究では、合成モデルとして GenSQL を用いて生成された合成データを対象に、元データ（テストデータ）との分布の類似性を TVD によって測定する。まず、元データおよび合成データそれぞれから共通する特徴量の組み合わせを抽出し、その組み合わせごとにカテゴリ

の出現頻度を集計する。次に、クロス集計表を作成し、得られた出現頻度を全体に対する割合として正規化することで、単純な行数比較ではなく、確率的な分布比較を可能とする。このとき、元データと合成データ間で整合しないカテゴリが存在する場合、それらについては相対頻度を 0 として補完を行い、両者が同一のカテゴリ空間上で比較可能な状態を整える。

このようにして得られた元データおよび合成データの相対頻度分布を用い、それぞれを $p(x)$ および $q(x)$ として TVD を算出する。具体的には、各カテゴリペア x に対して $|p(x) - q(x)|$ を求め、それらを総和した上で 2 で割ることにより TVD を導出する。TVD 値が小さい場合は、合成データが元データの分布特徴をよく再現していることを示し、逆に大きな値は分布に乖離があることを意味する。

本研究においてこの手続きで求められる TVD 値を分析することにより、GenSQL による合成モデルの分布再現性を定量的に評価できる。たとえば、データ数削減による影響や minutes パラメータ増加などの合成精度の改善策の有効性を判断する上で、TVD は重要な参考資料となる。これにより、合成モデルの品質を改善し、元データの特徴をより忠実に反映した合成データの生成を目指す上での指針が得られる。

3.4 minutes パラメータによる推論時間の制御

GenSQL の構造学習プロセスにおいて、推論に割り当てる時間を動的に制御することは、モデル精度の向上や計算資源の有効活用において重要な要素となる。ここで取り上げる minutes パラメータは、主に CGPM で使用される設定項目であり、ユーザはこれを通じて推論処理に費やす時間を明示的に指定できる。

ここで CGPM（Composable Generative Probabilistic Models）とは、確率生成モデルを構築・操作するためのフレームワーク。これは、データの生成、推論、条件付き分布の計算などを柔軟に行うために設計されている。

推論時間を増やすことで、モデルはより多くの探索と計算を行う余裕が生まれる。その結果、より複雑なデータ構造から潜在的なパターンをより的確に抽出し、モデル精度を高めることが可能となる。これにより、データセットの特性や分析の要件に応じて推論時間を適宜調整することで、最終的な分析結果の品質を最適化することが可能となる。

3.5 GenSQL の実験設定とパラメータ

本研究では、GenSQL を用いて合成データの生成および評価を行った。しかし、GenSQL のセットアップに関する詳細な手順については、本論文では割愛する。セットアップに関する情報は、公式ドキュメントを参照されたい。公式ドキュメント [13] には、ソフトウェアのインストール方法や基本的な使用法、必要な環境設定について詳しく記載されている。

以下では、GenSQL を用いた実験の具体的な手順について説明する。

まず、元データとして使用する data.csv ファイルを準備する。このデータには、実験で使用する全ての情報が含まれており、適切な前処理を施した上で、/GenSQL.structure-learning/data/ ディレクトリに配置する。このディレクトリ構成は、GenSQL が構築された環境内でデータを正確に認識させるために必要である。

次に、実験で使用するパラメータを設定するために、params.yaml ファイルを編集する。このファイルでは、データ数 N と学習にかける時間（minutes パラメータ）を指定する。データ数 N は実験の規模を、minutes はモデル学習にかける計算時間を制御する重要な要素である。

パラメータを設定した後、dvc repro コマンドを実行する。このコマンドは、dvc.yaml ファイルに記述されたパイプラインを順次実行し、モデルの学習と必要なデータ処理を行う。これにより、GenSQL が自動的にモデルを構築し、指定した条件に基づいてデータ生成を進める。

さらに、実験の評価に使用するテストデータとして、test.csv ファイルを準備し、/GenSQL.

structure-learning/data/test/ ディレクトリに配置する。このファイルには、元データの一部を用いて分離した検証用データが含まれており、合成データとの比較に利用される。

次に、生成された合成データ synthetic-data-gensql.csv とテストデータ test.csv を用いて、Total Variation Distance（TVD）を算出する。この処理は、compare-data.py を実行することで行い、2 つのデータセットの指定したカラム間の TVD を計算して、結果を tvd-data.json ファイルに出力する。

さらに、TVD 結果を視覚的に確認するための HTML ファイルを生成する。tvd-data.json の内容をテンプレートファイル/templates/fidelity.html に挿入する。

最後に、tvd-statistics.py を実行して、TVD 結果の平均値と標準偏差を算出する。このスクリプトにより、合成データとテストデータの分布一致度に関する詳細な統計的評価が可

能となる。

以上の手順を通じて、本研究では合成データの生成および評価を一貫して実施し、GenSQL の有効性を検証した。

3.5.1 使用したプログラミングのファイル

以下の compare-data.py で、全変動距離（TVD）を計算する。

```
import json
import os
import matplotlib.pyplot as plt
import pandas as pd

plt.rcParams["font.size"] = 14

data_means = json.load(open("data/data_means.json"))

df1 = pd.read_csv("data/test/test.csv")
df2 = pd.read_csv("data/synthetic-data-gensql.csv")

total_variation_distances = {}

# ターゲットのペアをループする
targets = list(data_means.keys())

for i in range(len(targets)):
    for j in range(i + 1, len(targets)):
        target1, target2 = targets[i], targets[j]

        # ターゲットがデータフレームに存在しない場合もスキップ
        if target1 not in df2.columns or target2 not in df2.columns:
            continue

        # 組み合わせごとの出現頻度をカウント
        count_file1 = df1.groupby([target1, target2]).size().unstack(
    fill_value=0)
        count_file2 = df2.groupby([target1, target2]).size().unstack(
    fill_value=0)

        # 出現頻度を比率に変換
        ratio_file1 = count_file1 / count_file1.values.sum()
        ratio_file2 = count_file2 / count_file2.values.sum()
```

```
        # 二つのカラムのすべての組み合わせに対して、欠損値にを埋める0
        ratio_file1 = ratio_file1.reindex(ratio_file2.index, fill_value=0).
    reindex(columns=ratio_file2.columns, fill_value=0)
        ratio_file2 = ratio_file2.reindex(ratio_file1.index, fill_value=0).
    reindex(columns=ratio_file1.columns, fill_value=0)

        # 全変動距離を計算
        total_variation_distance = (abs(ratio_file1 - ratio_file2)).values.
    sum() / 2
        total_variation_distances[f"{target1}, {target2}"] =
    total_variation_distance

# 全変動距離を低い順に表示
sorted_distances = sorted(total_variation_distances.items(), key=lambda x:
    x[1])
max_length = max(len(target) for target in total_variation_distances.keys()
    )
for target, distance in sorted_distances:
    print(f"{target.ljust(max_length)}: {distance:.4f}")

# Saving the TVD and column labels as a JSON file
json_data = [
    {"column-1": pair.split(', ')[0], "column-2": pair.split(', ')[1], "tvd
    ": distance, "model": "gensql", "index": idx}
    for idx, (pair, distance) in enumerate(sorted_distances)
]

with open('result/tvd_data.json', 'w') as json_file:
    json.dump(json_data, json_file, indent=4)
```

以下の tvd-statistics.py で TVD 結果の平均値と標準偏差を算出する。

```
import json
import csv
import statistics

# ファイルからデータを読み込むJSON
json_filename = 'result/tvd_data.json'
with open(json_filename, mode='r') as json_file:
    json_data = json.load(json_file)

# の値を抽出TVD
```

```
tvd_values = [entry['tvd'] for entry in json_data]

# 平均値と標準偏差を計算
mean_tvd = statistics.mean(tvd_values)
std_dev_tvd = statistics.stdev(tvd_values)

# ファイルに結果を書き込むCSV
csv_filename = 'result/tvd_statistics.csv'
with open(csv_filename, mode='w', newline='') as csv_file:
    csv_writer = csv.writer(csv_file)
    csv_writer.writerow(['Metric', 'Value'])
    csv_writer.writerow(['Mean␣TVD', mean_tvd])
    csv_writer.writerow(['Standard␣Deviation␣TVD', std_dev_tvd])
```

第4章　評価実験

4.1　実験設定

評価実験では、前述の提案手法を用いて、データ数を 150,000 行から 100 行まで段階的に削減するシナリオを設計する。具体的には、150,000 行、100,000 行、80,000 行、50,000 行、40,000 行、30,000 行、20,000 行、10,000 行、9,000 行、8,000 行、7,000 行、6,000 行、5,000 行、4,000 行、3,000 行、2,000 行、500 行、400 行、300 行、200 行、そして 100 行といった粒度で条件を設定し、それぞれのデータ数について、合成データを生成して TVD を算出する。この段階的な削減により、「どの程度データ数を減らしても分布再現性が維持されるのか」「どの閾値以下で急激に再現性が損なわれるのか」が明らかになる。

4.2　実験結果

4.2.1　データ数による結果への影響

4.1,4.2,4.3 に示されているように、データ数（N）に応じた TVD（Total Variation Distance）の平均値と標準偏差には顕著な傾向が見られる。本図では、横軸にデータ数（対数スケール）を、縦軸に TVD の平均値をプロットし、その周囲に標準偏差を表す 95%信頼区間を示している。

まず、多量データ（N が 10,000 以上）では、TVD の平均値はおおよそ 0.05〜0.1 に収まり、元データ分布を忠実に再現していることが確認できる（4.1,4.2,4.3 の右端部分）。また、標準偏差も 0.05〜0.09 程度と小さく、合成データ生成の試行間におけるばらつきが少なく、非常に安定した品質が得られている。この結果は、豊富なサンプルが統計的特徴を正確に把握し、GenSQL が分布構造を適切に学習・再現していることを示唆するものである。

次に、中程度のデータ（N が 10,000 から 1000）では、TVD の平均値が若干上昇し、約 0.06〜0.15 の範囲内となっている（4.1,4.2,4.3 の中央部分）。標準偏差もやや増加が見られるが、この範囲では依然として再現性は高く、分布再現性は許容できる水準にとどまってい

る。データ数が減少しても、10,000 程度であれば GenSQL の性能が大きく損なわれることはないと考えられる。

一方、少量データ（N が 1,000 以下）では、TVD の値が大幅に上昇する傾向が確認できる（4.1,4.2,4.3 の左端部分）。特に、N が 1,000 や 500、さらには 100 程度になると、TVD は 0.15〜0.3 以上の値に達し、元データ分布から大きく乖離した合成データが生成されることが明らかである。また、標準偏差も増大し、試行間のばらつきが顕著になっている。このような精度劣化の原因として、データ数不足により統計的特徴の推定が不安定になること、稀なカテゴリ値や領域が適切にカバーされないことが挙げられる。

4.1,4.2,4.3 の結果から、データ数が合成データの再現性や品質に大きな影響を与えることが示されている。特に、データ数が多いほど TVD は低く安定し、元データ分布を忠実に再現できる一方で、少量データ条件ではその精度が著しく低下する傾向が見られる。

図 4.1: minutes=1 の結果

4.2.2 カテゴリ数による結果への影響

データ内のカテゴリ数が TVD（Total Variation Distance）に与える影響について説明する 4.1,4.4。カテゴリ数が少ない列は、データ数が少ない場合でも分布を比較的正確に再現できる。一例として、性別（SEX）は「男性」と「女性」の 2 カテゴリのみを持つため、合

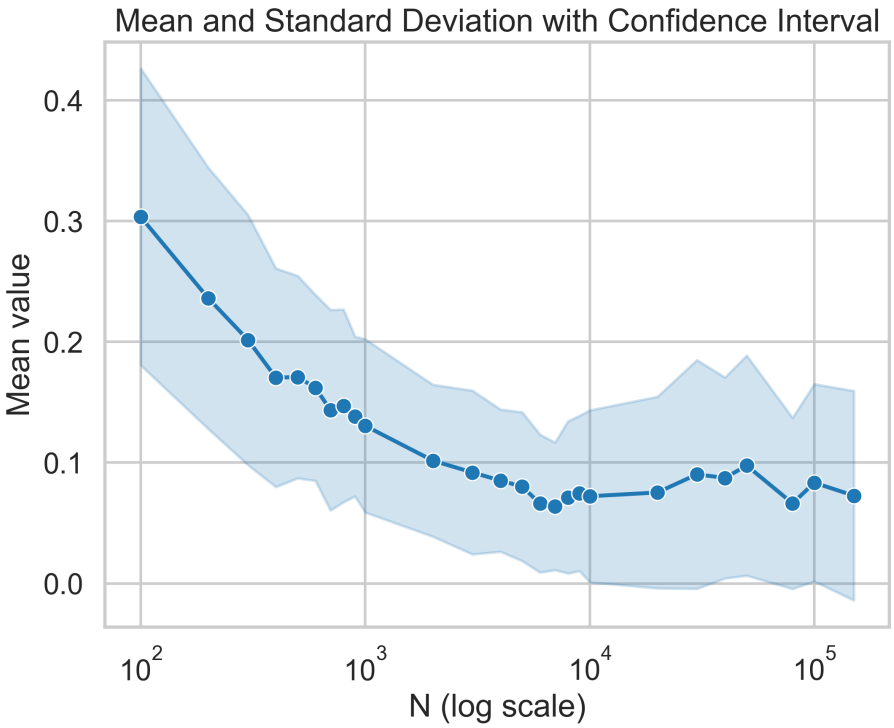

図 4.2: minutes=10 の結果

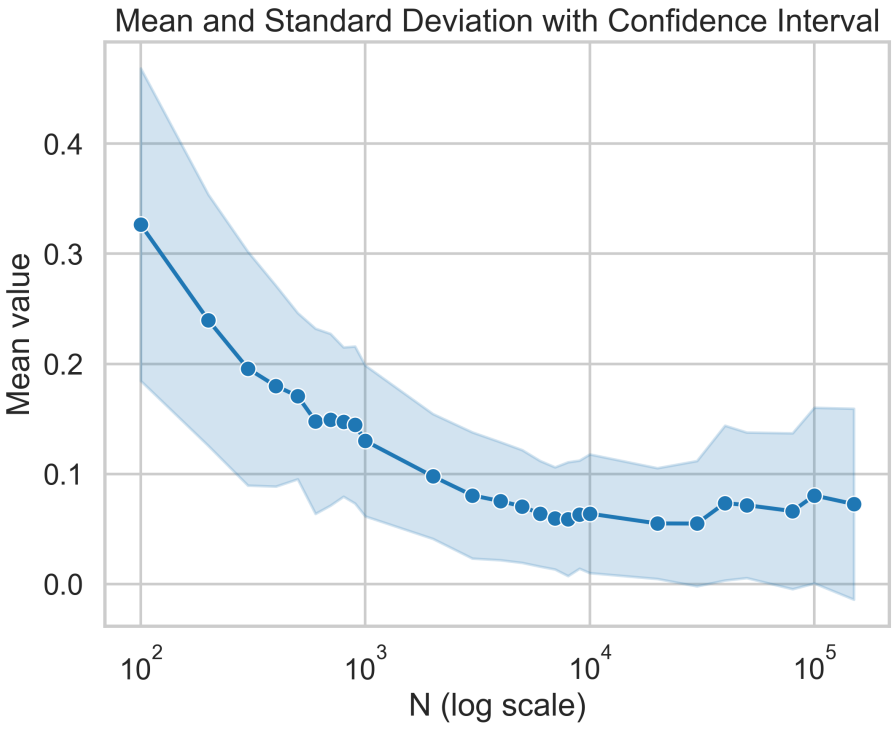

図 4.3: minutes=60 の結果

成データにおいても各カテゴリの出現頻度を安定して学習できる。このような特徴量では、TVD が低い値に抑えられる傾向がある。

一方、カテゴリ数が中程度の列では、データ数が不足すると低頻度のカテゴリの分布が歪んだり、完全に消失したりする可能性がある。たとえば、年代別区分（AGEID）は 11 カテゴリ、居住地域（AREA）は 8 カテゴリを持つが、これらの列では低頻度カテゴリの学習が難しくなり、TVD が増加する要因となる。

さらに、カテゴリ数が多い列では、データ数の不足がより深刻な影響を及ぼす。たとえば、所属・関連する産業分野（Q1）は 31 カテゴリを持つため、各カテゴリに十分なデータが行き渡らない状況が生じやすい。その結果、低頻度カテゴリの出現頻度が偏ったり、完全に消えてしまうことがある。このような列では、TVD が大幅に上昇する傾向が見られる。

また、カテゴリ数が中程度でも分布の偏りが顕著な列では注意が必要である。たとえば、テレビ接触頻度（Q3S1）や 1 日あたりのテレビ視聴時間（Q4S1）は、それぞれ 8 カテゴリを持つが、特定のカテゴリに回答が集中しやすい。この場合、低頻度カテゴリの分布が正確に再現されない可能性が高く、TVD の上昇を引き起こすことがある。

なお、カテゴリ数と TVD には傾向としての相関が見られるものの、データ数の影響により、TVD の正確な大小関係が必ずしもカテゴリ数順に一致するわけではない点にも留意する必要がある。たとえば、カテゴリ数が多い列でも十分なデータ量があれば TVD が抑えられる場合や、カテゴリ数が少なくても分布が極端に偏っている場合には TVD が高くなることがある。それでも、一般的な傾向として、カテゴリ数が多いほど学習の難易度が上がり、データ数が限られている場合には TVD が高くなる傾向がある。

以上のように、カテゴリ数が多いほど学習の難易度が上がり、データ数が限られている場合には TVD が高くなる傾向が明確である。一方、カテゴリ数が少ない列では分布の再現性が高く、TVD が低い値に抑えられる。このことから、データのカテゴリ数と学習精度の関係性を考慮することが重要である。

4.2.3 minutes パラメータの影響

本研究では、顧客データを用いて生成される合成データの精度に対するデータ数の影響を分析するにあたり、minutes パラメータを変更することでデータ生成時間の異なる条件下での性能を評価した。その結果、minutes=1 4.1, minutes=10 4.2, minutes=60 4.3 の条件それぞれにおいて、TVD や信頼区間に顕著な違いが見られた。

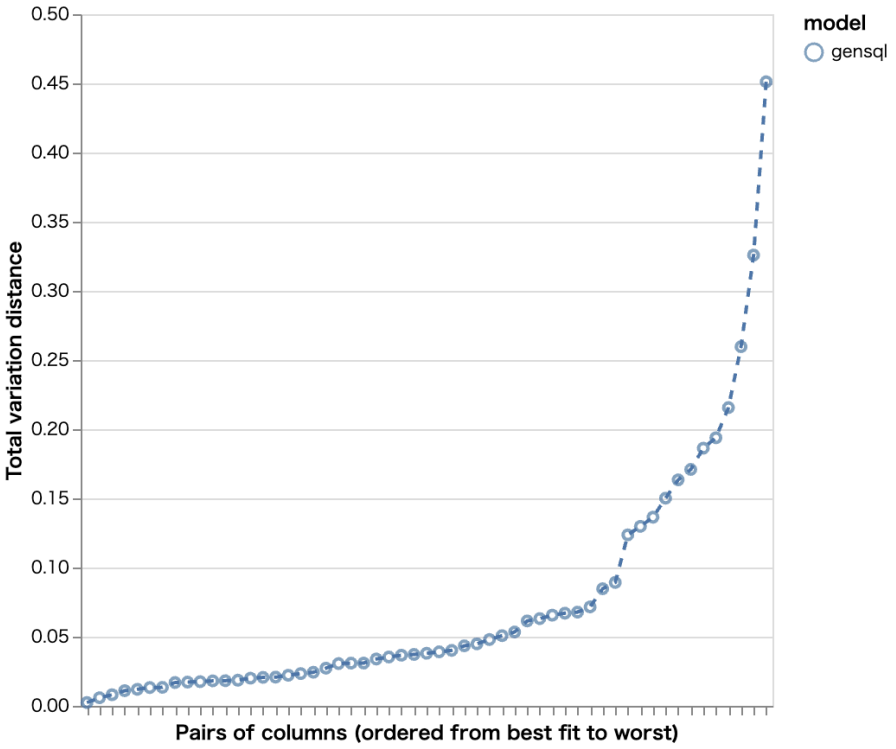

図 4.4: N=150,000、minutes=60 での column のペアの TVD

多量データ（10,000 行以上）の場合、生成時間が短いほど分布一致度が低下し、信頼区間が広がる傾向が確認された。特に minutes=1 では、TVD が他の条件に比べて高く、生成データの安定性に欠けることが示された。一方で、minutes=10 および minutes=60 では信頼区間が狭まり、安定性が向上することが分かった。ただし、データ数が十分に多いため、minutes=10 の条件でも精度がほぼ確保される結果となった。

中程度のデータ（1000 行から 10,000 行）においても、生成時間が短いほど分布一致度が低下し、信頼区間が広がる傾向が見られた。この条件では、特に minutes=60 が最も安定した結果を示し、分布一致度が他の条件を大きく上回った。一方で、minutes=10 では一定の精度が確保されているものの、minutes=1 では信頼区間が広く、TVD が低下するなど、精度面での限界が見られた。

少量データ（1,000 行以下）では、いずれの minutes 条件においても分布一致度や信頼区間に大きな変化は見られなかった。データ数が非常に限られているため、生成時間による影響が相対的に小さくなる傾向が示された。この結果から、少量データ条件では生成時間を増やしても分布一致度や信頼性に大きな改善が見られないことが示唆された。

表 4.1: N=150,000、minutes=60 での column のペアの TVD

column-1	column-2	TVD	column-1	column-2	TVD
SEX	MARRIED	0.00239	AGEID	CHILD	0.03907
MARRIED	Q3S1	0.00588	HINCOME	Q4S1	0.04017
CHILD	Q3S1	0.00801	AREA	HINCOME	0.04339
AREA	MARRIED	0.01095	PINCOME	Q4S1	0.04483
MARRIED	PINCOME	0.01196	AREA	JOB	0.04795
AREA	Q3S1	0.01324	MARRIED	CHILD	0.05075
SEX	CHILD	0.01337	SEX	HINCOME	0.05340
SEX	Q3S1	0.01687	JOB	Q4S1	0.06138
CHILD	HINCOME	0.01718	Q1	Q4S1	0.06304
PINCOME	Q3S1	0.01752	AGEID	Q4S1	0.06556
CHILD	PINCOME	0.01807	AREA	Q1	0.06698
AREA	CHILD	0.01819	SEX	AGEID	0.06775
AGEID	MARRIED	0.01857	AGEID	Q3S1	0.07156
SEX	AREA	0.02003	AGEID	PINCOME	0.08467
MARRIED	Q4S1	0.02058	AGEID	HINCOME	0.08923
CHILD	Q4S1	0.02077	HINCOME	Q1	0.12362
AREA	Q4S1	0.02214	HINCOME	JOB	0.12976
HINCOME	Q3S1	0.02334	AGEID	Q1	0.13639
SEX	Q4S1	0.02430	AGEID	JOB	0.15011
MARRIED	JOB	0.02724	SEX	Q1	0.16338
AGEID	AREA	0.03058	SEX	PINCOME	0.17091
CHILD	JOB	0.03089	SEX	JOB	0.18628
MARRIED	Q1	0.03093	Q3S1	Q4S1	0.19382
CHILD	Q1	0.03387	PINCOME	Q1	0.21561
AREA	PINCOME	0.03533	PINCOME	JOB	0.25966
JOB	Q3S1	0.03661	HINCOME	PINCOME	0.32559
MARRIED	HINCOME	0.03717	JOB	Q1	0.45119
Q1	Q3S1	0.03802			

4.3 考察

はじめに、データ数の大小が合成データの品質、とりわけ TVD（Total Variation Distance）に大きく影響することが明らかとなった。具体的には、10,000 行以上の十分なデータ数が得られる場合、合成データとの TVD は 0.05〜0.1 に抑えられ、標準偏差も小さく安定していた。この結果は、元データが豊富であれば、GenSQL を用いて元データとほとんど遜色のない分布特性を再現可能であることを示唆している。一方、1,000 行を下回るような極端にデータ数が少ない条件下では、TVD が 0.1〜0.3 と大きく増加し、標準偏差も拡大した。これは、出現頻度の低いカテゴリや特徴量の分布を十分に学習できず、分布の類似性が大きく損なわれるためである。

また、カテゴリ数が多い列や列数自体が多い状況では、特に少量データ（N が 1,000 以下）で出現頻度の低いカテゴリを正確に捉えることが難しく、TVD の上昇が顕著に表れる。このようなケースでは、カテゴリを必要以上に細分化しないこと、あるいは十分なデータ数と学習時間を確保することが、合成データにおいて分布を正確に再現するうえで非常に重要となる。

ここでは、minutes パラメータが顧客データを用いた合成データ生成に与える影響について検討した。多量データ条件および中程度のデータ条件では、生成時間を短く設定すると信頼区間が広がり、TVD が低下する傾向が確認された。例えば、minutes=1 では他の条件と比べて分布一致度が著しく低く、信頼区間も広がるため、生成データの精度と安定性に課題があることがわかった。一方、minutes=60 では最も高い精度が得られ、信頼区間も狭く、安定したデータ生成が可能である。また、minutes=10 に設定すると minutes=1 に比べて TVD が大きく向上し、効率性と精度のバランスを保つ実用的な選択肢となる。一方、少量データ条件では minutes パラメータの影響が相対的に小さく、いずれの設定でも TVD や信頼区間に顕著な差は見られなかった。これは、データ数が限られている場合に生成時間を延ばしても精度が飛躍的に向上しにくいことを示している。

最後に、総括すると、多量データ（10,000 行以上）や中程度のデータ（1000 行から 10,000 行）を扱う際には、minutes=60 の設定によって精度と安定性が最も高い合成データを得られるが、minutes=10 でも精度と効率を両立可能であるといえる。一方、少量データ（1,000 行未満）の場合は、元々の分布を正確に再現することが難しく、さらに minutes パラメータによる改善効果も限定的であった。そのため、限られた条件下では、生成時間を短縮することが妥当な判断となる場合がある。以上より、合成データを生成する際には、データ数を十分に確保することが品質向上の鍵であり、データ数に応じて minutes パラメータを適切に調整する必要があると結論づけられる。

4.4 今後の展開

今後の展開としては、まず評価指標や分析手法の拡張が考えられる。本研究では TVD によって分布比較を行ったが、これに加えて異なる距離・類似度尺度を導入すれば、より多面的な分布評価が可能になる。また、純粋な分布再現性だけでなく、合成データを購買予測モデルの訓練や顧客セグメンテーション、リスクアセスメントなどのタスクに用いた際のモデル性能を比較することで、合成データが実務に耐える代替リソースとなり得るかを評価する

タスクベースの手法も有望である。

次に、合成モデル自体の改良や新技術の投入も視野に入る。本研究では GenSQL を用いたが、少量データ下でも分布情報を効率的に取り込めるモデルや手法へ拡張することが考えられる。その一例として、強い事前情報（ドメインナレッジ）を組み込むことで、データ不足による不確実性を軽減する方法が挙げられる。

分散環境やプライバシー保護技術との融合も重要な方向性である。複数の組織がデータを共有せずに学習を行うフェデレーションラーニング環境下では、元データに直接アクセスすることなく各拠点で合成データをローカル生成し、分析者間で共有する仕組みが有効になり得る。また、合成データ生成プロセスに差分プライバシー技術を組み込むことで、個人情報を保護しつつ全体的な分布特徴を再現した安全性の高い合成データが作成でき、データ提供者や利用者の双方にとって利点が大きい。

さらに、本研究で示した手法や知見は、ユーザ顧客データ以外の領域にも応用可能である。医療分野の電子カルテや診療履歴、金融取引データ、あるいは IoT 機器から得られる時系列センサーデータなど、さまざまなドメインで分布再現性を検証することで、提案手法の一般性と限界を明らかにできる。その結果として、より汎用的なガイドラインを構築することが可能となる。

最後に、実務導入を円滑に行うために、データ数や minutes パラメータといった条件と得られる TVD や精度改善の程度を関連づけた実践的な指標の整備が挙げられる。実務担当者が合成データ生成技術を導入する際に、どの程度のデータ数や minutes パラメータが必要なのか、どれほどの精度改善が期待できるのかを、標準化されたプロトコルやベストプラクティスとして提示することで、合成データ生成の実用性はさらに高まると考えられる。

第5章　まとめ

これまでの実験結果を総合的に振り返ると、データ数と合成データの精度には明確な関係性があることが確認された。まず、多量データでは、合成データ生成手法である GenSQL が元データの分布特性を忠実に再現できることが示された。たとえば、データ数が多量データ（データ数が 10,00 以上）では、合成データと元データ間の TVD（Total Variation Distance）は 0.05〜0.1 に収まり、標準偏差も比較的低い水準であった。これは、元データの統計的特徴をモデルが十分に捉え、分布特性を安定して再現できることを示唆している。こうした条件下では、合成データは元データに非常に近い分布的整合性を持ち、代替データとして十分に活用可能である。

さらに、多量データ（データ数が 10,00 以上）および中程度のデータ（データ数が 10,000 から 1,000）においては、合成データ生成の際に設定する minutes パラメータが結果に与える影響も無視できないことがわかった。具体的には、生成時間を短く設定すると信頼区間が広がる一方で、TVD が低下する傾向が見られたが、minutes=1 のような極端に短い場合には、TVD が低くなり、信頼区間も広がってしまうため、生成データの精度と安定性に課題が生じていた。一方、minutes=60 では最も高い精度が得られ、信頼区間も狭く、安定したデータ生成が可能であることが確認された。また、minutes=10 に設定すると minutes=1 よりは TVD が大きく向上し、効率性と精度のバランスを保つ点で実用的な選択肢となる。

一方、データ数を段階的に削減した結果、少量データ（データ数が 1,000 以下）では合成精度が著しく低下することが明らかになった。特にデータ数が 1,000 未満になると、TVD は 0.1 を超え、500 や 100 といった極端に少ない条件では 0.2〜0.3 台に達した。さらに、標準偏差が増大し、試行ごとのばらつきが大きくなる傾向も確認された。この現象は、モデルがデータ数不足のため分布特性を正確に推定できず、出現頻度の低いカテゴリや特徴範囲を適切に再現するための情報が不足していることに起因していると考えられる。こうした少量データでは minutes パラメータの調整による影響が相対的に小さく、生成時間を延ばしても精度を飛躍的に向上させることは難しいことも分かった。

今後の課題としては、少量データからでも高品質な合成データを生成するためのモデル改

善が挙げられる。さらに、評価指標の多様化やタスクベースの実践的評価を通じて、合成データの品質を多角的に検証する必要がある。また、フェデレーションラーニングや差分プライバシーといった新しい技術との統合、異なるデータ種類や応用分野での再検証を進めることで、合成データ生成技術の汎用性と実用性をさらに向上させることが期待される。

参考文献

[1] Mathieu Huot, Matin Ghavami, Alexander K. Lew, Ulrich Schächtle, Cameron E. Freer, Zane Shelby, Martin C. Rinard, Feras A. Saad, and Vikash K. Mansinghka. Gensql: A probabilistic programming system for querying generative models of database tables. In Proceedings of the ACM Programming Languages (PLDI), pp. Article 179, 26 pages, June 2024.

[2] Cynthia Dwork and Aaron Roth. Algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, Vol. 9, No. 3-4, pp. 211–407, 2014.

[3] Marco F. Cusumano-Towner, Feras A. Saad, Alexander K. Lew, and Vikash K. Mansinghka. Gen: A general-purpose probabilistic programming system with programmable inference. In Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '19), pp. 22–26. ACM, 2019.

[4] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. arXiv preprint arXiv:1907.00503, 2019. 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

[5] Noah D. Goodman and Andreas Stuhlmüller. Probabilistic programming: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 38, No. 1, pp. 20–38, 2014.

[6] Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus A. Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. Journal of Statistical Software, Vol. 76, No. 1, pp. 1–32, January 2017.

[7] Anand Patil, David Huard, and Christopher J. Fonnesbeck. Pymc: Bayesian stochastic modelling in python. arXiv, Vol. 1810.09538, pp. 1–81, 2010.

[8] Joshua V. Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A. Saurous. Tensorflow distributions, 2017. Available at https://www.tensorflow.org/probability/api_docs/python/tfp/distributions.

[9] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. Bayesian Data Analysis. CRC Press, 3rd edition, 2013.

[10] Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

[11] Christian P. Robert and George Casella. Monte Carlo Statistical Methods. Springer, 2004.

[12] Radford M. Neal. Probabilistic inference using markov chain monte carlo methods. Technical report crg-tr-93-1, University of Toronto, 1993.

[13] GenSQL Documentation. Quick start - structure learning, 2025. Accessed: 2025-01-11.

英語要約

English Title

Data Volume and Accuracy Evaluation in Synthetic Data Generation Using GenSQL

It has been confirmed that there is a clear relationship between the number of data points and the accuracy of the generated synthetic data. First, for large datasets, it was demonstrated that the synthetic data generation method called GenSQL can faithfully reproduce the distributional characteristics of the original data. For instance, under large-scale conditions of over 10,000 data points, the TVD (Total Variation Distance) between the synthetic data and the original data remained roughly within the 0.05–0.08 range, and the standard deviation was relatively low. This suggests that the model is sufficiently capturing the statistical properties of the original data and can stably replicate their distributional characteristics. Under these conditions, the synthetic data exhibits very close distributional consistency with the original data and can be fully utilized as a substitute dataset.

Furthermore, for both large and medium-sized datasets, it was found that the minutes parameter set during synthetic data generation has a non-negligible impact on the results. Specifically, while setting a shorter generation time tended to expand the confidence interval, it also tended to reduce the TVD. However, in extreme cases such as minutes=1, although the TVD was lower, the confidence interval became wider, leading to challenges in the accuracy and stability of the generated data. On the other hand, with minutes=60, the highest accuracy was achieved, and the confidence interval was narrow, enabling stable data generation. Additionally, setting minutes=10 significantly improved

the TVD compared to minutes=1, making it a practical choice for balancing efficiency and accuracy.

On the other hand, when the dataset size was reduced step by step, it became clear that the synthesis accuracy declined markedly for small datasets. In particular, when the number of data points fell below 1,000, the TVD exceeded 0.1; under extremely small conditions, such as 500 or 100 data points, the TVD reached the 0.2–0.3 range. Moreover, the standard deviation increased, indicating a greater variability in results across trials. This phenomenon is likely due to the model ' s inability to accurately estimate distributional characteristics with insufficient data, lacking the information necessary to properly replicate less frequent categories and feature ranges. It was also found that, for such small datasets, adjusting the *minutes* parameter had a relatively minor effect and that extending the generation time did not dramatically improve accuracy.

Future challenges include improving the model so that even small datasets can yield high-quality synthetic data. Additionally, there is a need to diversify evaluation metrics and conduct task-based practical assessments to examine the quality of synthetic data from multiple perspectives. Moreover, integrating new technologies such as federated learning and differential privacy, as well as revalidating the approach across different data types and application areas, is expected to further enhance the generality and practicality of synthetic data generation technologies.

謝辞

最後に、ご指導していただいた指導教官の岡瑞起准教授に深く感謝いたします。また、沢山の助言をいただいた研究室の皆様にお礼申し上げます。特に岩橋七海さん、吉田祥平さんには大変お世話になりました。本当にありがとうございました。