

小口研究室 研究紹介 (2023年度)

(お茶の水女子大学理学部情報科学科)

都営バスのオープンデータを用いた渋滞検知に向けた不均衡データ対応手法の適用と評価 (研究担当: 畠中 希)

研究背景

渋滞による被害

- ・ 経済的損失約12兆円
- ・ 時間的損失約30時間/人
- ・ 停車車両の排気ガスによる大気汚染
- ・ 救急車等の緊急車両の到着遅延

渋滞検知方法と問題点

- ・ 方法1: 道路上の感知器
→ 感知器が無い道路では渋滞検知が不可
- ・ 方法2: 渋滞予想サービス
→ ユーザの通行データが少ない道路では予測精度が低くなる可能性がある

先行研究

都営バスのリアルタイム運行データを用いた渋滞検知^{*1}

感知器のない道路も走行・定期的に運行しているため十分なデータの収集が可能



▶ 学習データが不均衡なためaccuracyに比べrecallやf1-scoreが低い

^{*1} 青柳宏紀, 岡田一光, 山名早人. 都営バスのリアルタイム運行データを用いた渋滞検知. DEIM2022 第14回データ工学と情報マネジメントに関するフォーラム, 2022, pp.1-8
^{*2} 渋滞とは時速10km以下で継続して走行する状態

提案手法

不均衡データに対する3つのアプローチ

サンプリングアプローチ

すでにあるデータからランダムにデータを選び複製する

コストアプローチ

それぞれのクラスにコストを与えて学習する

ハイブリッドアプローチ

サンプリングアプローチ後にコストアプローチを行う

評価実験

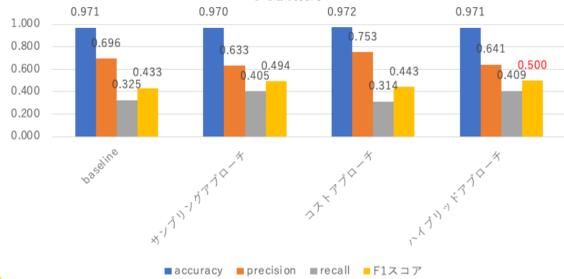
実験

日時: 2022年11月30日~12月13日, 12月18日~12月31日
対象路線: 都02, 池65, 早77, 高71, 門19, 平23
データ数: 106002個 (渋滞 3402個, 97220個)
アルゴリズム: ランダムフォレスト

特徴量

- ・ 停留所区間のバスの走行速度
- ・ 同一停留所区間走行時の1つ前のバスの速度
- ・ バス走行時の時間帯
- ・ 区間終点停留所の出発時刻と定刻との差
- ・ 区間始点停留所の出発時刻と定刻との差

実験結果



- ・ baselineは不均衡データへのアプローチを行わず、ランダムフォレストのみを行った手法
- ・ 全てのアプローチがbaselineよりF1スコアが高くなった
- ・ **ハイブリッドアプローチ**が最もF1スコアが高い結果となった

- ・ データを増やしたことで特徴が掴みやすくなった
- ・ 少数派のデータを重要視するようになったため予測精度が上がった

まとめと今後の展望

まとめ

- ・ 停留所区間ごとに「渋滞」「非渋滞」の二値分類を行った
- ・ 不均衡データへ3つのアプローチを行った
- ・ ハイブリッドアプローチが最も高い結果になった

今後の展望

- ・ 新たな特徴量を追加
- ・ ディープラーニングの使用を検討

量子回路シミュレータの実行環境変化に伴う性能評価と特性比較 (研究担当: 青木 望美)

研究背景と目的

近年注目されている量子コンピュータは、一部の問題について従来型コンピュータと比較して非常に高速な計算を実現可能であることから盛んに研究・開発が行われている。しかし、理想的な挙動をする量子コンピュータの実現には数年から数十年かかると言われている。ここで量子シミュレータというものが存在する。量子シミュレータは従来型コンピュータ上で量子コンピュータの挙動を表現するツールであり、量子コンピュータ上で動作する量子アプリ等の開発に有用である。本研究では複数の量子シミュレータの分析を行い、性能特性の比較や高速化に向けた検討を行う。

実験概要

分析対象の量子シミュレータにQiskit AerとQulacs, mpiQulacsを用いる。また、用いる量子シミュレータはState Vector方式のものであり、State Vector方式のシミュレータは結果の測定前の量子状態を取得することが可能なことから、量子アプリ等開発時のデバッグ作業に有用である。分析対象の量子回路は、ランダム回路であるQV回路とシーケンシャルなQSB回路を使用する。また、量子ビット数は30、深さは10とする。これらの量子回路を実行し、複数の性能分析ツールを使用して性能データを収集する。

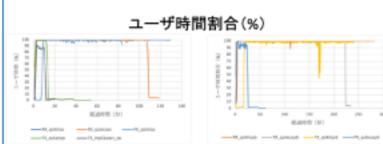
サーバ	Primergy RX2540 M1	サーバ	FX700
OS	Rocky Linux 8.6	OS	Rocky Linux 8.5
CPU	Intel Xeon プロセッサ E5-2697v3(2.60GHz)	CPU	Fujitsu A64FX (2.0GHz)
CPUコア数	14	CPUコア数	48
L1d/L2/L3 キャッシュ	32 KB(core)/256 KB(core)/35 MB	L1d/L2 キャッシュ	64KB(core)/8MBx4(CMG)
メモリ容量	128 GB	メモリ容量	HBM2 32 GB

サーバ	Primergy RX2540 M1	サーバ	FX700
OS	Rocky Linux 8.6	OS	Rocky Linux 8.5
CPU	Intel Xeon プロセッサ E5-2697v3(2.60GHz)	CPU	Fujitsu A64FX (2.0GHz)
CPUコア数	14	CPUコア数	48
L1d/L2/L3 キャッシュ	32 KB(core)/256 KB(core)/35 MB	L1d/L2 キャッシュ	64KB(core)/8MBx4(CMG)
メモリ容量	128 GB	メモリ容量	HBM2 32 GB

実験結果

量子回路実行時間(秒)

QV回路	RX	FX700	QSB回路	RX	FX700
Qiskit	129.1	77.5	Qiskit	280.7	223.5
Qulacs	119.0	49.5	Qulacs	233.8	57.0
mpiQulacs	-	19.5	mpiQulacs	-	-



キャッシュミス率(%)

QV回路				QSB回路			
RXサーバ	L1D	L2	L3	RXサーバ	L1D	L2	L3
Qiskit	1.5	45.9	67.9	Qiskit	2.1	39.1	58.9
Qulacs	2.7	54.3	38.4	Qulacs	2.4	53.0	37.6
FX700サーバ	L1D	L2	FX700サーバ	L1D	L2		
Qiskit	1.5	29.6	Qiskit Aer	2.8	16.5		
Qulacs	2.8	34.2	Qulacs	3.2	42.7		
mpiQulacs	3.8	56.1	mpiQulacs	-	-		



まとめ

量子回路実行時間はQiskit Aer > Qulacs > mpiQulacs、QulacsとmpiQulacsはサンプリング処理の実行時間が比較的目的立つ。また、量子回路実行時間が短いシミュレータほどメモリバンド幅を引き出している傾向があった。したがって、今後はこれらの点を改善点として、具体的な性能向上案を検討していく。

辞書を用いた圧縮センシングによる準同型暗号文の通信量削減 (研究担当: 泉 湖雪)

研究背景

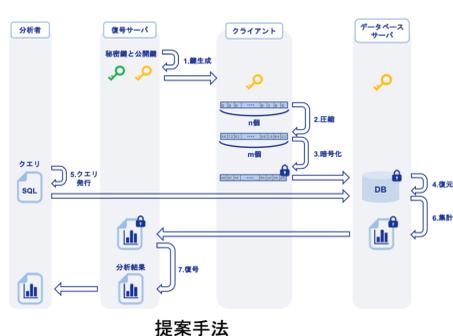
- ・ クラウドサービスへの外部からの攻撃によるデータの漏洩に備える
- ・ 個人情報を保護しながらデータ分析をしたい
→ 準同型暗号の使用が適している
- ・ 準同型暗号
暗号文同士で加算や乗算ができる暗号
暗号文サイズが大きいため通信量が多い
→ 多次元データに対して圧縮センシングを使用することでクライアント・サーバ間の通信量を小さくする

多次元データのサーバ送信時における通信量を削減する

- ・ 圧縮センシング
スパース性(ゼロ成分が多いという性質)を持つデータを圧縮してから復元する方法
- ・ 辞書学習
一部のデータから最適なスパース変換を学習すること

実験概要

- ・ **Baseline**
スパースな多次元データをクライアントで準同型暗号化して、圧縮せずにサーバに送信する
- ・ **提案手法**
辞書と圧縮センシングを利用して通信量を削減して多次元データを送信する
- ・ **実験環境**
クライアント: Raspberry Pi
サーバ: Ubuntuのマシン
- ・ **再構成アルゴリズム**
 $encrypt(\hat{x}) = D((AD)^T(AD))^{-1}(AD^T) \times encrypt(y)$
辞書Dを用いて一般逆行列と暗号化された低次元行列yとの積を求めると復元する



実験結果

- ・ 暗号文のサイズ
提案手法では Baseline に比べて約半分の通信量に抑えられており、通信量の削減率は n と m の比に対応している
- ・ 実行時間
暗号文のサイズの減少に伴い暗号化にかかる時間は減っているが、サーバで復元する際の再構成アルゴリズムの計算に時間がかかっている
n, m の値が小さいほどクライアントでの圧縮時間やサーバでの復元時間が短くなっている
- ・ 精度
学習データが多くても圧縮率が高いと精度が下がる、即ち圧縮率のほうで精度に与える影響が大きい
圧縮率 50% (n = 256, m = 128), 学習量 25% としたときの 256 個の x の RMSE は、おおよそ 0.025 となった
また、圧縮率 50% において、Baseline, 提案手法で各カラムごとにカウントクエリを実行した結果も示す

