

# 長大な数値系列を扱うための LLM の外部メモリ機構

浅野 輝<sup>1,2</sup> 大内 啓樹<sup>3,4,2</sup> 春日 瑛<sup>4</sup> 米谷 竜<sup>4</sup>

<sup>1</sup> 東京大学大学院 <sup>2</sup> 理化学研究所

<sup>3</sup> 奈良先端科学技術大学院大学 <sup>4</sup> サイバーエージェント

asano-hikaru19@g.ecc.u-tokyo.ac.jp, hiroki.ouchi@is.naist.jp

{kasuga\_akira, yonetani\_ryo}@cyberagent.co.jp

## 概要

大規模言語モデル (LLM) に長大な数値時系列をそのまま入力すると、計算コストの増大に加えて推論品質の低下が生じやすい。本研究では、数値時系列に対して省トークンかつ説明可能な参照・推論を可能にする外部メモリ機構を提案する。提案手法は二段階からなり、第一に Bayesian Online Changepoint Detection (BOCPD) により時系列をオンラインにセグメント化し、各セグメントに統計量と自然言語要約を付与して逐次格納する。第二に、クエリに応じて関連セグメントを抽出し、それらを束ねるメタセグメントをツール駆動で動的に構築することで、必要な粒度の情報提示と段階的推論を実現する。人工的な日次売上データを用いた実験では、年度別トレンド分析において LLM が適切にツールを使用し、根拠となる統計量を参照しながら最終回答を生成できることを示した。

## 1 はじめに

大規模言語モデル (LLM) が現実の意思決定や分析業務に組み込まれるにつれて、長期的な文脈を保持し必要時に取り出す「メモリ」の設計が重要な研究課題となっている。長い入力列をそのまま与えると計算コストが急増し、重要情報が埋もれて推論精度が劣化しやすい。そこで、外部メモリや検索を組み込む RAG に加え、グラフ [1] や木 [2]、タイムライン [3] といった構造で記憶を整理する手法が提案され、複雑な質問応答や長い対話における性能向上が報告されてきた。しかし、こうした成功の多くは、自然言語テキストのように「意味単位」が比較的明確に切り出せるデータを前提としている。

一方で、実務における重要情報の多くは、売上やセンサ値のような数値データとして長大な時系列の形で蓄積される。既存のメモリ手法は、段落・発話

といった明示的な境界や語彙に基づく意味の類似性など、言語的手掛かりを前提として設計されており、これらが乏しい数値時系列には直接適用しにくい。数値時系列では、生成過程の切り替わりが連続的に生じ得るため「どこで区切るか」が本質的に曖昧であり、「近さ」の定義も形状・振幅・周期性など複数の観点に分解され、どの観点が支配的かは目的に依存する。以上の理由から、数値時系列に対する LLM 外部メモリは設計の自由度が大きい一方で失敗の仕方も多様であり、汎用的な設計原理は未だ確立していない。

そこで本研究では、数値時系列向け LLM 外部メモリを提案する。本手法は、数理的な連続性に基づいて「セグメント」という最小単位を確定する第一段階と、クエリに応じてそれらを束ねる「メタセグメント」を構築する第二段階からなる。第一段階では、Bayesian Online Changepoint Detection (BOCPD) [4, 5] により時系列を統計的に安定なセグメント列へ分割し、各セグメントに統計量と自然言語サマリを付与して保存する。これにより、生データを直接扱わず、意味のある最小単位として参照可能にする。第二段階では、セグメント列に対してクエリに応じた動的な構造化を行う。本手法では、複数のセグメントを束ねて要約する「メタセグメント」を導入する。個々のセグメントは、時間的なまとまり (例: 月次集約) や性質的なまとまり (例: 急落区間の集合) など、複数のメタセグメントに同時に所属できる。LLM はツールを介して関連セグメントを検索し、メタセグメントをオンデマンドに生成・キャッシュする。この多重所属を許容する構成により、LLM は問いの粒度や観点到合わせて時系列を柔軟に再構成し、大域的な傾向把握と局所的な詳細参照を両立した推論が可能となる。

本研究では、提案手法が長期系列に対しても適切な分析が可能であることを実験的に検証した。

## 2 準備

本節では、本研究で扱う数値時系列の逐次入力形式、オンライン変化点検出 (BOCPD) の最小限の表記、ならびに後段の動的グラフ構築で用いるセグメント、メタセグメントとグラフの用語を整理する。

### 2.1 逐次入力される数値時系列とセグメント記憶

本研究が対象とするデータは、時間とともに逐次到着する数値時系列である。簡単のため観測値は1次元とし、観測時刻を  $t = 1, 2, \dots$ 、観測値を  $x_t \in \mathbb{R}$  と書く。実務上では、一定の時間区間に含まれる複数の観測がまとめて到着することが多い。そこで本研究では、到着ステップを  $n = 1, 2, \dots$  とし、ステップ  $n$  で時間区間  $I_n \subset \mathbb{N}$  に属する観測がバッチ

$$B_n = \{x_t\}_{t \in I_n} \quad (1)$$

として入力される状況を考える。例えば「1日ごとに、その日の毎時の気温が送信される」場合、 $n$  は日を表し、 $I_n$  は当該日の時刻インデックス集合に対応する。

LLM が扱えるコンテキスト長には制約があり、逐次到着する長大な生データ  $\{B_n\}$  を毎回すべてプロンプトへ投入する設計は計算コストの面で現実的でない。さらに、系列が長くなるほど LLM の性能が劣化することも報告されている [6]。そこで本研究では、数値データを直接保持せず、時系列をオンラインに区間分割して得られる**セグメント**をメモリの最小単位として蓄積し、検索・要約の対象とする。

セグメント  $s_k$  は時間区間  $[t_k^s, t_k^e]$  ( $t_k^s \leq t_k^e$ ) で表され、その内部の観測列を  $X_k = \{x_t \mid t_k^s \leq t \leq t_k^e\}$  と書く。セグメント境界は入力バッチの境界と一般に一致しないため、1つのセグメントが複数のバッチにまたがる場合や、1つのバッチが複数のセグメントに分割される場合がある。

時刻  $t$  までに確定し格納されたセグメントの集合を  $\mathcal{S}_t = \{s_k\}_{k=1}^{K_t}$  とし、各セグメント  $s_k$  は少なくとも (i) 区間  $[t_k^s, t_k^e]$ 、(ii) 統計量 (特徴量) ベクトル  $\phi_k$ 、(iii) 自然言語サマリ  $\sigma_k$  を属性として持つものとする。以降では、生データ列ではなく  $\mathcal{S}_t$  に対する操作を考える。

### 2.2 オンライン変化点検出

変化点検出は、時系列の生成過程がある時点で切り替わることを検知し、区間を分割する問題である。

本研究では、逐次到着するデータに対してセグメント境界をオンラインに更新する、Bayesian Online Changepoint Detection (BOCPD) [4, 5] を用いる。

BOCPD では、ランレングス  $r_t$  を「時刻  $t$  において直近の変化点からの経過ステップ数」を表す変数として導入する。変化点の発生はハザード関数  $H(\cdot)$  により制御され、 $H(r_{t-1})$  が大きいほど変化点が発生しやすくなる。BOCPD は各時刻でランレングスの事後分布を逐次更新し、変化点確率を計算する。具体的な更新式や観測モデルの選択、ハザード関数の設計などの詳細は既存文献 [4, 5] を参照されたい。

本研究の関心は変化点検出そのものではなく、長大時系列を統計的に安定な最小単位 (セグメント) へ分解してメモリ化する点にある。そのため、BOCPD により得られる変化点確率に基づいて境界を確定し、確定した区間を順次  $X_k$  として閉じ、 $\phi_k, \sigma_k$  を付与して  $\mathcal{S}_t$  に格納する。

### 2.3 LLM 外部メモリのグラフ表現

自然言語テキストを対象とする LLM 外部メモリの多くは、会話の切れ目や段落、文書チャンクといった「ひとまとまりの文章」を記憶単位として切り出し、外部メモリに格納する。推論時には、入力クエリに対して関連する記憶単位を検索し、上位候補をコンテキストとして LLM に提示することで、長期文脈を省トークンで参照する。

これらのメモリ設計では、記憶単位同士の関係を構造として保持することで、頑健な検索と推論を実現する試みが広く採用されている。代表例が近傍グラフであり、各記憶単位をノードとして扱い、意味的に近い記憶単位間にリンクを張ることで近傍構造を形成する。このリンクは話題の連なりを表し、グラフ上の局所探索により、単発の類似検索では拾いにくい補助情報や根拠を段階的に回収することを可能にする。

さらに近年は、メモリに木構造 (あるいは階層グラフ) を導入し、葉ノードに原子的な記憶単位を、内部ノードにその部分集合の要約を保持させるなど、より効率的なメモリシステムが提案されている。これにより、粗い粒度の要約から大域的な文脈を捉え、必要に応じて詳細を参照する段階的な推論が可能となる。

### 3 提案手法

本節では、逐次到着する数値時系列に対して、LLM が省トークンで参照・推論可能な外部メモリ機構を定式化する。提案手法は二段階からなる。第一に、BOCPD により時系列をオンラインでセグメント化し、各セグメントに統計量と自然言語要約を付与して逐次格納する。第二に、クエリ  $q$  に応じて関連セグメントを抽出し、それらを束ねるメタセグメントと部分グラフをオンデマンドに構築することで、推論に必要な情報の提示を可能にする。

#### 3.1 オンラインセグメント化とメモリへの逐次格納

到着ステップ  $n$  でバッチ  $B_n = \{x_t \mid t \in I_n\}$  が入力されるたびに、BOCPD によりランレングス事後分布  $p(r_t \mid x_{1:t})$  を逐次更新し、得られる変化点確率に基づいてセグメント境界をオンラインで確定する。境界が確定した区間  $[t_k^s, t_k^e]$  に対して観測列

$$X_k = \{x_t \mid t_k^s \leq t \leq t_k^e\}$$

を閉じ、統計量ベクトル  $\phi_k$  と自然言語サマリ  $\sigma_k$  を付与して

$$s_k = ([t_k^s, t_k^e], \phi_k, \sigma_k)$$

としてメモリに追加する。時刻  $t$  までに確定・格納されたセグメント集合を  $\mathcal{S}_t = \{s_k\}_{k=1}^{K_t}$  とし、以降の検索・構造化・推論は生データではなく  $\mathcal{S}_t$  上の操作として記述する。

#### 3.2 動的なメタセグメント構築

数値時系列では、「どの区間が関連するか」および「どの粒度で要約すべきか」がクエリに強く依存して変化する。例えば、月次の推移を問うクエリと、急落イベントの連鎖を問うクエリでは、注目すべき区間も要約粒度も一致しない。このため本研究では、事前に固定的な類似度や階層構造を与えるのではなく、セグメントを最小単位として確定した上で、クエリに応じて上位単位である**メタセグメント**を動的に定義する。

具体的には、時刻  $t$  までに確定したセグメント集合  $\mathcal{S}_t$  から、クエリ  $q$  に関連する候補セグメントを LLM によるツールの使用を通して抽出し、それらを束ねたメタセグメントを構成する。LLM は、メタセグメントの要約を起点に全体像を把握し、必要に応じて下位のセグメントへ掘り下げることで、省トークンかつ説明可能な段階的推論を実現する。

### 3.3 ツール駆動による動的メタセグメント構築

**基本設計** 本研究では、候補抽出・集約・要約・保存を LLM のツールユースとして実装する。クエリ  $q$  が与えられると、LLM は (i) 関連セグメントの検索、(ii) メタセグメントの構成、(iii) 要約の付与、(iv) 必要に応じた追加検索・再要約、を反復し、最終的な回答を生成する。重要なのは、クエリごとに常に全てを再構築するのではなく、一度生成したメタセグメントを保存して再利用し、必要な場合にのみ増補する運用を採用する点である。

**候補セグメント集合の抽出** メタセグメント構築の出発点として、LLM はまず候補セグメント集合を抽出する。候補抽出は、時間条件（例：直近 1 ヶ月、特定期間）、値域条件、統計量条件（例：分散や傾きの閾値）に基づくフィルタリングを基本とし、必要に応じて自然言語サマリ  $\sigma_k$  を手がかりに関連セグメントを追加する。これにより、クエリに対して十分に関連しつつ、後段の要約・推論に適したサイズの候補集合を得る。

**メタセグメントの定義：多重所属と再帰的集約** メタセグメント  $m$  は、複数のセグメント（あるいは下位メタセグメント）を束ねた上位単位であり、その構成要素の集合を  $C(m)$  と書く。セグメントは単一のまとまりに固定されず、複数のメタセグメントに同時に属してよい。例えば、同一のセグメントが「2026 年 1 月」という時間的なまとまりにも、「高ボラティリティ区間」という性質的なまとまりにも属し得る。この多重所属を許すことで、数値時系列における多面的な切り口（時間粒度、値域、イベント種別、場所・デバイス等の属性）を無理なく表現できる。

さらに、**メタセグメントを束ねた上位メタセグメント**も生成できる。例えば、日次のまとまりを週次へ、週次を月次へ、月次を四半期・年次へと段階的に集約することで、時間的階層性を自然に導入できる。同様に、店舗 → 地域 → 都道府県のような階層も同一の枠組みで扱える。LLM はクエリの要求粒度に応じて適切なメタセグメントを生成・参照し、必要なら下位へ降りて個々のセグメントを確認することで、長期トレンドと局所変動を統一的に扱う。

各メタセグメントには、構成要素の統計量を集約した代表値（例：平均・分散・極値・トレンド）と、構成要素のサマリを統合した自然言語サマリを付与する。これにより、LLM はメタセグメントの要約

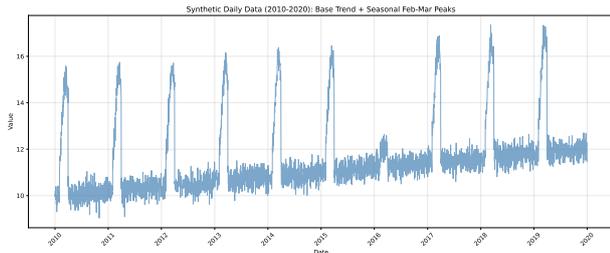


図1 実験結果のプロット

から全体像を把握し、根拠が必要な箇所だけをセグメントへ掘り下げる段階的推論を実行できる。

**再利用と再現可能性** LLM が実行した候補抽出・メタセグメント生成・要約付与の結果は、ツール入出力としてログ化し、生成されたメタセグメントをキャッシュする。これにより、類似クエリに対する再利用が可能になると同時に、「どのセグメント集合をどの粒度で束ね、その要約に基づいて回答したか」を追跡でき、分析業務に必要な再現性と説明可能性を担保できる。

## 4 実験

### 4.1 データセット

本実験では、2010年1月から2020年12月までの人工的な日次売上データを作成した。全体的に上昇傾向を示し、2月の売上が上がる季節性を設定した。ただし、2016年のみトレンドの増加幅が小さくなるように設定した。観測期間は2010/01/01から2020/12/31までであり、系列長は約4018件である。

データ生成は、(i) 長期トレンド、(ii) 月次の季節性、(iii) 年ごとのトレンド強度の違い、(iv) 観測ノイズ、から構成した。これにより、年単位の比較(2016年の鈍化)と月単位の比較(2月の押し上げ)を同一系列上で同時に問えるようにしている。

### 4.2 定性評価

以下に、今回の提案手法におけるツールユースの例を示す。本タスクでは、LLM に対して yearly trend の分析を依頼した。その結果、LLM は適切にツールを使用して、最終的な回答を行なっていることが確認できる。図2にツール使用の履歴とエージェントによる分析結果を示す。

**タスク定義** 本節で示すクエリは「年ごとの傾向 (yearly trend) を要約せよ」というものであり、プロンプトには、ツールの説明とともに、タスクの定義

#### ツール使用の履歴

```
Query: Yearly trend analysis
[Step 1/10] tool: ['time_bounds']
[Step 2/10] tool: ['create_meta_segment_by_datetime_range']
[Step 3/10] tool: ['get_meta_features']
[Step 4/10] tool: ['get_meta_features']
[Step 5/10] tool: ['get_meta_features']
[Agent] Final answer at step 6/10
```

#### エージェントによる分析結果

年ごとの日次売上データの傾き (slope\_mean) は以下の通り：

年	slope_mean
2010	-0.00023
2011	-0.00023
2012	-0.00272
2013	-0.00091
2014	-0.00508
2015	+0.00673
2016	+0.00030
2017	-0.00570
2018	-0.00437
2019	-0.00060

The data shows mostly slight negative trends in sales for most years, with the exception of 2015 and 2016 which show positive or near flat trends. The magnitude of slopes is generally small, indicating gradual changes in sales over the years.

図2 LLM エージェントによるツールの使用と分析結果を記述した。

図2では、LLM がまず分析対象期間を確定し (time\_bounds)、次に年単位のまとまりをメタセグメントとして定義し (create\_meta\_segment\_by\_datetime\_range)、最後に各年の集約特徴量を取得して (get\_meta\_features) 結論を述べている。このように、計算 (集約・傾き推定) と説明 (言語化) を分離することで、長大な生データの提示を避けつつ、根拠となる統計量を明示した回答を生成できる。

## 5 おわりに

本研究では、長大な数値時系列に対して LLM が省トークンで参照・推論できる外部メモリ機構を提案した。提案手法は、BOCPD によるオンラインセグメント化でセグメントを確定し統計量と要約を付与する段階と、クエリに応じてメタセグメントを動的に構築し必要な粒度で情報を提示する段階からなる。人工的な日次売上データを用いた定性評価により、LLM がツールを用いて年次傾向を分析し、根拠となる特徴量を参照しながら回答を生成できることを示した。今後の課題として、実データでの定量評価、クエリ種別ごとの最適な要約粒度・特徴量設計の検討、ツール入出力ログを用いた再現性評価の体系化が挙げられる。

## 参考文献

- [1] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitan, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2025.
- [2] Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for LLMs. In **The Thirteenth International Conference on Learning Representations**, 2025.
- [3] Kai Tzu-iunn Ong, Namyoun Kim, Minju Gwak, Hyungjoo Chae, Taeyoon Kwon, Yohan Jo, Seung-won Hwang, Dongha Lee, and Jinyoung Yeo. Towards lifelong dialogue agents via timeline-based memory management. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, **Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, pp. 8631–8661, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.
- [4] Ryan Prescott Adams and David J C MacKay. Bayesian online changepoint detection. **arXiv [stat.ML]**, October 2007.
- [5] Matias Altamirano, Francois-Xavier Briol, and Jeremias Knoblauch. Robust and scalable bayesian online changepoint detection. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, **Proceedings of the 40th International Conference on Machine Learning**, Vol. 202 of **Proceedings of Machine Learning Research**, pp. 642–663. PMLR, 2023.
- [6] Asano Hikaru, Ouchi Hiroki, Kasuga Akira, and Yonetani Ryo. MobQA: A benchmark dataset for semantic understanding of human mobility data through question answering. **arXiv [cs.CL]**, August 2025.

## A 付録

### A.1 実験設定の補足

本研究の実験では、gpt-4.1-mini<sup>1)</sup>を使用した。モデルへの入力プロンプトには、タスク定義に加えて、対象データ（時系列の概要、期間、観測間隔、想定される分析観点など）の説明を与えた。また、外部メモリに対する操作を行うために、利用可能なツール群（各ツールの役割、入力引数、戻り値の形式）を併記し、モデルが必要に応じてツールを選択できるようにした。

ツール呼び出しの際には、LLM が所定のフォーマットに従って、使用したいツール名とツールへ入力する引数をそれぞれ出力するように設計した。システム側では、この出力からツール名と引数を抽出して実行し、得られた結果を再びモデルへ提示することで、逐次的なツールユースを可能にした。この手続きにより、長大な生データを直接プロンプトへ埋め込むことなく、必要な統計量や集約結果を根拠として参照しながら、最終的な回答を生成できる。

---

1) <https://platform.openai.com/docs/models/gpt-4.1>