

# マルコフ転換モデルによる仮想通貨市場の分析

井澤 公彦\*

## I 序論

### 1 はじめに

#### 1.1 この研究の意義

現在日本では貨幣決済から、キャッシュレス決済に移行するべきであるという風潮が高まっている。永井 [2018] では日本の決済手段の現状について、「2015年現在，世界各国のキャッシュレス決済比率の比較を行うと，キャッシュレス化が進んでいる国は40%～60%台であるのに対し，日本は約20%にとどまっている」と述べており，現金決済比率の高さについて問題提起している。電子決済，クレジットカード，デビットカードなどがキャッシュレス決済の手段としてあげられるが，貨幣決済にとって代わるもう一つの候補になりうるのがビットコインを筆頭とする仮想通貨である。ビットコインは，Nakamoto [2008] によって提案されたモデルをもとに2009年から運営が始まり，CoinMarketCap [2019] のデータによるとその時価総額は2015年1月1日には42億9753万5768円しかなかったものが2017年12月17日には最大で35兆5743億9755万1220円まで高騰し，4年で約8000倍に価値が膨らんだ。その後ビットコインの価格は下落したものの2019年11月では時価総額は15兆円ほどで推移しており依然存在感を見せている。

一方で仮想通貨には様々なリスクが存在する。

まず，仮想通貨は政府が関与しない通貨であることから，安全性が取引所に依存してしまうという問題がある。例えば2018年にコインチェックが保有する仮想通貨のNEMがほぼ100%流出するという事件が起こり，仮想通貨の価値の信頼性や安全性が失われたことから通貨価格が大幅に下落した。また，2014年には顧客が保有する75万ビットコインと自社保有の10万ビットコインがシステムバグを悪用した不正アクセスにより盗まれ消失したことからビットコイン取引所を運営していたMTGOXが破綻した。

他にも，詳細は2.2や2.3で説明するが，ビットコインを含む仮想通貨はボラティリティが従来の貨幣と比べて高く，価値の貯蔵に適さないことや，ビットコインネットワークがビットコインの需要に耐えられず決済手数料が高騰してしまい決済機能としてのメリットが失われてしまうなどの問題点がある。

このように仮想通貨に対しては，ビットコインを中心に新たな決済手段として期待されている。しかし一方で，ビットコインのリスクの多さから，現金決済にとって代わるには至っていないのが現状である。

---

\* 京都大学大学院経済学研究科修士課程

本研究では、仮想通貨の貨幣機能としての問題点とその影響を分析するために、仮想通貨の時系列データを基にした定量分析を行う。仮想通貨の定量分析の意義は、価格変動の因果関係や構造を分析することによって、仮想通貨の特徴を明らかにできることである。本研究では特に取引コストに注目することで、ビットコインの貨幣機能の現状とその影響を明らかにすることを旨とする。このため、仮想通貨が決済手段としての現状を論じる上で本研究は有意義なものである。

## 1.2 本研究の目的と研究方法

本研究の目的は、仮想通貨の貨幣機能の問題点が価格にどのような影響を及ぼしているかについて定量的に分析することである。本研究では、特に、ビットコインの決済機能の問題点を分析し、それがどのように価格に影響を及ぼすかを検証する。具体的には、トランザクション手数料やメモリープールの変動率と、ビットコインの価格変動率がどのような関係にあるかを分析する。なお、本研究では二段階の検証を行う。最初に、線形回帰分析を行い、データ間の関係を調査する。次に、トランザクション手数料、メモリープールを情報変数としてマルコフ転換モデルを活用したデータ分析を行う。マルコフ転換モデルを使うメリットについて、沖本 [2010] (p181) では、「経済・ファイナンスのデータは景気や投資家の心理など、観測できない変数に影響を受けるものが多く、MSモデルは観測できない変数の状態によって特性が異なるデータを分析するのに便利である」と述べている。

本研究においては Diebold, Lee and Weinbach [1994] にて発案されたモデルを使用する。この論文によると、従来のマルコフ転換モデルは周期的なレジームを感知するのに効果的である一方で、推移確率が一定になるという制約がある。これはつまり、一つのレジームからもう一つのレジームに転換するときの確率をその状況下の経済ファンダメンタルズに依存させたモデルを作ることができないということである。一方、このモデルでは推移確率に情報変数を入れることによって経済ファンダメンタルズに依存させることができる。本研究では、仮想通貨の情報が、ビットコインにどのような影響を与えたかを分析する際には、ビットコインのアメリカドルに対する価格変動率の平均と分散をモデルのパラメーターとする。そしてトランザクション手数料、メモリープールの外部情報がレジームの転換にそれぞれどのような影響を及ぼすか、もしくはその情報がどのようにビットコイン価格の平均と分散を二つのレジームに分離するかを観測する。本研究においてはビットコイン価格については CoinMarketCap [2019] のビットコインのアメリカドルに対する価格の日時データを用いる。また、トランザクション手数料、メモリープールのデータは Blockchain [2019]<sup>1)</sup>のデータを参照する。これ以降登場する図形やグラフは Matlab2018a のソフトを活用したものである。今回のマルコフ転換モデルを活用する際には、データの対数差分系列を利用し、情報となるデータについても対数差分系列を用いる。今回使用する、ビットコインの時系列データと対数差分データは以下の図のようになる。なお期間は 2017 年 10 月 16 日から 2019 年 10 月 15 日までの 2 年間である。

---

1) メモリープールのデータについて 2 点ほどデータの変更を行った。1 点目としてはメモリープールのデータは約 21 時間ごとのデータであり、1 日に 2 つデータがある日はその平均をその日の値にした。2 点目として 2019 年 1 月 31 日にメモリープールが 0 というデータがあったが異常値なのが明らかであったため 1 月 30 日と 2 月 1 日のデータの平均を使った。

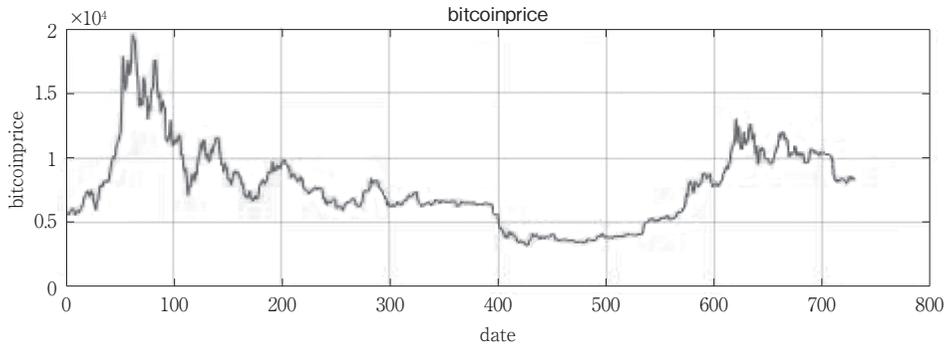


図1 ビットコインの価格の時系列データ

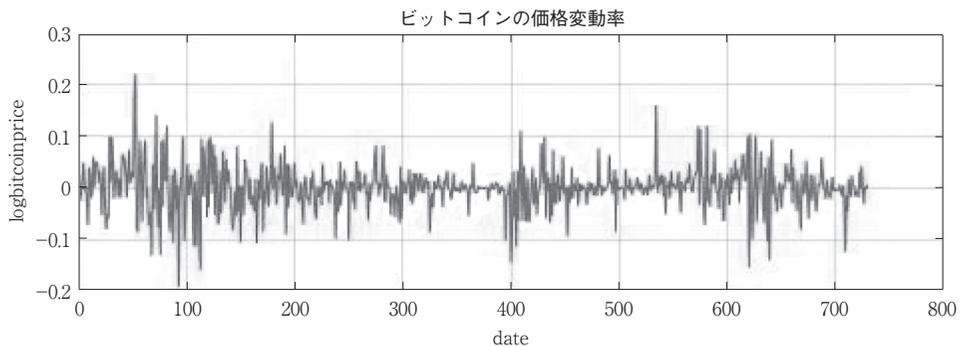


図2 ビットコインの価格変動率の時系列データ

## 2 ビットコインの意義と問題点

### 2.1 ビットコインの意義

現状の決済システムについて Nakamoto [2008] では以下の問題点を挙げている。それは、現状のインターネット上の取引システムは第三者の信用に依存しているということである。第三者が仲介することの問題点は取引の仲介手数料が高くつくことである。例えばクレジットカード取引や銀行振り込みに多額の手数料が必要である。一方で、Nakamoto [2008] では第三者の信用が必要な理由として二重支払い問題<sup>2)</sup>をあげている。つまり、手数料を安く抑えるためには、第三者の仲介なしにこの問題を解決することができる決済システムが必要であり、それが可能であれば仲介手数料による取引の機会損失を防ぐことができるということである。この論文ではビットコインのシステムである“peer-to-peer distributed timestamp”を使うことで、第三者の信用に頼ることなく、二重支払い問題を解決した。

### 2.2 ビットコインのボラティリティ

次にビットコインの問題点について述べる。

1.1でも述べたように、ビットコインには以下の3つの問題点がある。

2) 一度支払われたはずのお金が再び使用されうるとの問題

- ①ビットコインのボラティリティが高い。
- ②取引所の信頼性が低い。
- ③ビットコインが普及すると取引コストが上昇する。

この節ではこれらの問題点のうち①について解説する  
 経済学には一般的に貨幣には3つの機能があると定められている。

堀, 二神 [2017] によると, 貨幣には, 価値の基準, 交換手段, 価値の貯蔵手段, の三つの機能がある。このうち価値の基準や価値の貯蔵手段としての役割を果たすには貨幣としての価値の安定性が不可欠である。そのためビットコインを貨幣として評価するためには, ビットコインの価値の安定性について検証することが必要である。

ここでは, ビットコインが従来の貨幣と比べてどれくらいボラティリティが高いのかを検証する。具体的には日本円とビットコインの貨幣としての価値の安定性を比較する。検証方法としては米ドルとの交換レートの変遷の対数差分系列の比較を行う。具体的には, ビットコインの対数差分系列と日本円の対数差分系列を比較した。日本円のデータは, DataStream を用いた。この比較を行った結果, 図3のような結果を得ることができた。

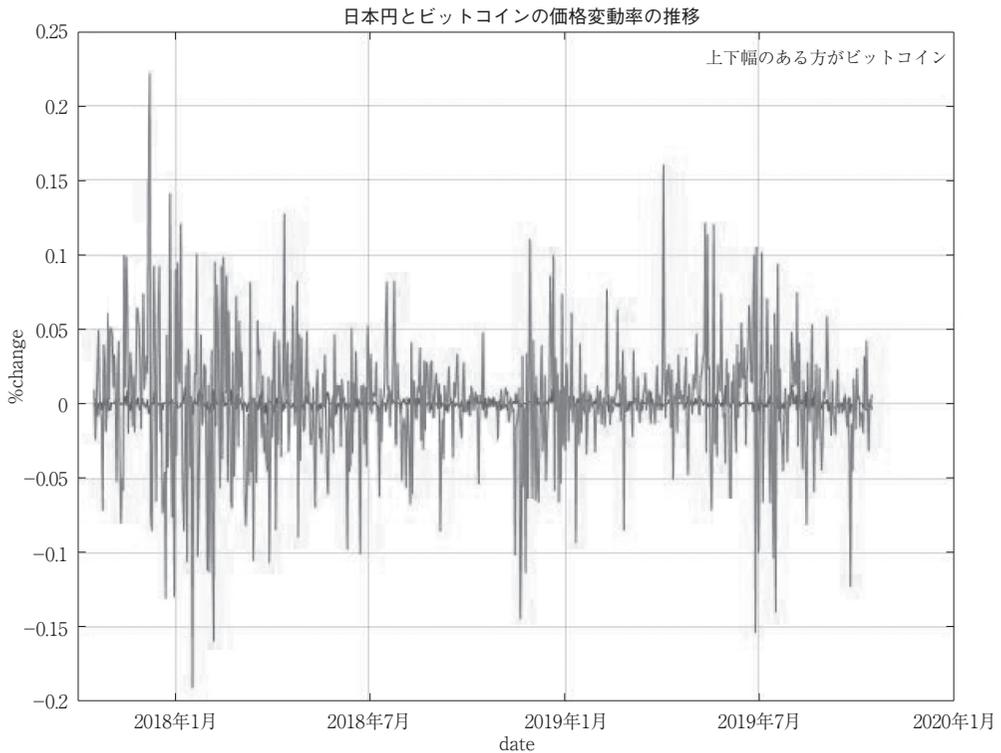


図3 日本円の価格変動率の時系列データ

図3より, ビットコインは日本円と比べて, 貨幣価値の変化が明らかに大きく, 価値基準や価値の貯蔵手段としての適性が低いことが分かる。

### 2.3 ビットコインの取引コストの高騰

次にビットコインの普及が取引コストの上昇を引き起こす問題について説明する。ビットコインが貨幣の三技能の一つである交換手段としての役割を果たすためには、取引コストが低いことが重要である。しかし、ビットコインの取引コストはビットコインの普及とともに上昇した。その原因について、トランザクション手数料の決定方法、ビットコインの普及とトランザクション手数料の関係、メモリープールについての説明という三つの視点で解説する。なおトランザクションの仕組みについては字数の都合上割愛する。

まず、ビットコインのトランザクション手数料の決定方法について解説する。ビットコインのトランザクション手数料はトランザクションの構造のデータの大きさによって決定する。つまり、トランザクション手数料はデータの大きさ×1バイト当たりのトランザクション手数料によって決定される。ここからは、1バイト当たりのトランザクション手数料がどのように決定するかについて説明する。

ビットコインの1バイト当たりのトランザクション手数料は2つのステップを経て決定する。まず、需要側が1バイトあたりのトランザクション手数料を自由に決定する。次に、マイナーが送金側のトランザクション手数料を見て自分が引き受けたいトランザクションを見つけマイニングに参加する。つまり送金側は早く決済を承認してほしい場合、トランザクション手数料を引き上げ、マイニングを行うインセンティブを引き上げ、送金時間が長くてもトランザクション手数料を抑えたい場合は、マイニングを行うインセンティブが低くてもいいので、トランザクション手数料を低めに設定する。

次にビットコインの普及がなぜトランザクション手数料の上昇を促したかについて説明する。ビットコインのトランザクションに使えるメモリは1回のブロックにつき1MBという制約がある。これはビットコインのスケーラビリティ問題と呼ばれ、Atomic Wallet [2019]によると、ビットコインのトランザクションに使えるメモリが少ないため、取引量や送金需要が高くなると、供給側が1MBの制限により供給が追いつかず、ビットコインのトランザクション手数料が高騰するという問題が発生する。この問題に対する対策として、Segwitというトランザクションの容量を節約して使う方法や、Bitcoin Cash<sup>3)</sup>という他の仮想通貨とハードフォークすることなどが行われた。しかし、これらの対策ではビットコインの需要の増加を吸収できず、図5のように、ビットコインの知名度が向上し、ビットコインの価格が高騰した2017年末から、2018年初頭にかけて、トランザクション手数料は爆発的に増加した。

最後にビットコインの取引コストを論じる上で重要な位置を占めるメモリープールについて解説する。Antonopoulos [2015] 今井, 鳩貝訳 [2016] (p166)によると、メモリープール<sup>4)</sup>は、「未検証のトランザクションの一時的なリスト」としている。そして、メモリープールにあるトランザクションはマイニングが行われブロックに収められるまでこのプールに待機することになるとされている。つまり、トランザクション手数料を低く設定したときには長くこのメモリープールにいる可能性が高くなることになる。

このメモリープールについても図4のように2017年末から2018年初頭の価格高騰の時にメモ

3) BitcoinCashは2017年にビットコインから分離(ハードフォーク)したものだ。

4) Antonopoulos [2015] 今井, 鳩貝訳 [2016] ではトランザクションプールという表記がされている。

リープールの値が増加し、ビットコインの普及が多くのトランザクションの承認を滞らせたことが分かる。

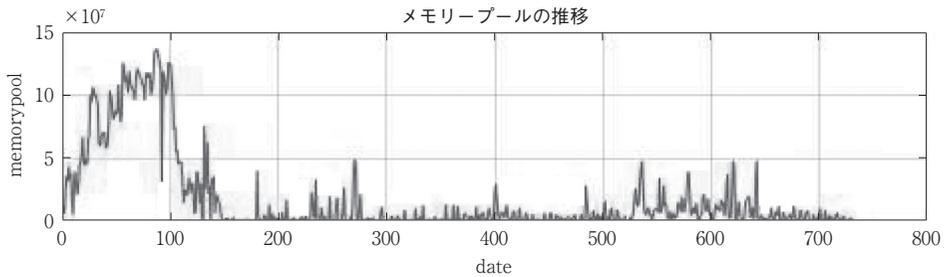


図4 メモリープールの量に関する時系列データ (2017/10/16~2019/10/15)

以上のことからビットコインの普及が取引コストの上昇を引き起こすことが分かる。本論では、取引コストの変化がビットコインの価格にどのような影響をもたらすかを分析する。

### 3 ビットコインの価格モデルに関する先行研究

仮想通貨は2節でも述べたように、為替と比較して非常にボラティリティの高い。それゆえに、仮想通貨の時系列分析に関する先行研究は非常に多く存在し、仮想通貨の価格モデルは日々研究されている。この節では、仮想通貨の価格モデルについて、いくつかのパターンに分けて先行研究を紹介する。

#### 3.1 ビットコインの認知度やインターネット上での評価を考慮した研究

ビットコインの価格モデル、もしくは価格の予想を行うときに、多くの研究が、SNSの評価やインターネット上の情報を活用している。例えば Matta, Lushesu, and Marchesi [2015] は “Senti Strength tool” という感情分析方法や、オプションマイニングの方法を用いて、2015年1月から3月の肯定的、中立的、否定的の三つに分類したツイッターの投稿と Google の検索トレンドの四つの情報と、ビットコイン価格の相互相関を調査した。その結果ビットコインに対して肯定的なツイートとビットコインの価格には、3日のラグを経て-0.35の負の相関関係を持つこと、Google の検索トレンドとビットコインの価格には0.64の強い相関関係があることを示した。また、Li, and Wang [2016] は、ビットコインの価格についてマイニングの難しさや経済状況のほかに Google での検索数、ツイート数も活用した ADLM (Autoregressive Distributed Lag Model) で分析を行った。その結果、Google での検索数に関しては3日のラグがあるとき正の強い影響が観測され、Twitter の検索に関しては5日のラグがあるとき強い負の影響があることが観測された。一方で、ツイッターやグーグルの検索数などの仮想通貨に対する反応全体を対象にするのではなく、より仮想通貨について専門的なコミュニティについて分析したものもある。Kim et al [2016] では、インターネット上の暗号通貨のコミュニティ<sup>5)</sup>の情報を「クローリング<sup>6)</sup>」という手法で収集し、コミュニティ内でのコメントやそれに対する返信の反応と、ビットコイン、イーサリアム、リップ

5) <https://bitcointalk.org/> のサイト内のコミュニティ

ルの価格の影響を、グレンジャー因果性<sup>7)</sup>を用いて分析した。その結果、ビットコインの価格については6日のラグがあるときコミュニティーで得られる情報を使うと価格予想において79%の精度で的中したほか、トランザクションの量においては3日のラグがあるとき77%の精度で的中した。また、イーサリアム及びリップルの価格においても6日、もしくは7日のラグがあるとき最も精度の高い価格予想をすることができたとしている。

### 3.2 従来のMSモデルを用いた研究

ビットコインの価格に対してMSモデルを用いて行った研究もある。Chappell [2018] ではビットコインの価格を2つから7つまでのレジームに分けてマルコフ転換モデルの推定を行い、ベイジアン (BIC), Hannan-Quinn (HQ), 赤池情報基準 (AIC) の三つの方法を用いて評価した。AICの評価では5つのレジーム、-HQCの結果では4つのレジームの時、BICのときは3つのレジームのモデルがそれぞれ適切であると評価した。また、分散の面を考慮すると、5つのレジームの時間が1番有効なモデルであることを示した。Ardia, Bluteau, and Rüede [2019] ではビットコインのボラティリティの高さに注目し仮想通貨の価格を、MSGARCHモデルを用いて分析し、それをVaR (Value at Risk) の手法を用いて評価を行った。

### 3.3 トランザクション手数料、メモリープールに関する研究

メモリープールもしくはトランザクション手数料に関する研究は2016年までは少なく、たとえ存在していたとしても機械学習の教師データの一つとしての扱いがほとんどである。例えばSin, and Wang [2017] では、memory poolなどを含んだデータを利用しニューラルネットワークモデルを構築することにより53%ベンチマークより上回るトレード戦略を作り上げた。

### 3.4 先行研究から見える課題

ビットコインに関する先行研究の一部を取り上げたが、ビットコインの価格モデルに関する論文は非常に多い。しかし、ビットコインに対して時系列分析を行うもののほとんどは如何にベンチマークに対して勝つかもしくは、知名度や経済状況がどのような効果を及ぼすかを分析することにとどまっており、ビットコインの決済システムの問題点に関係する定量分析を行っている論文はほとんどない。この点において、ビットコインの決済機能としての現状とビットコインの価格の関係を分析する行為は新規性がある。

## 4 本研究の流れ

本研究の目的は、ビットコインの取引コストの変化がビットコインの価格にどのような影響を与えるかを解き明かすことにある。本論では最初に情報変数に使用するトランザクション手数料とメ

---

6) クローリングとは、ロボット型検索エンジンにおいて、プログラムがインターネット上のリンクを辿ってWebサイトを巡回し、Webページ上の情報を複製・保存することである。(IT用語辞典より引用)

7) 現在と過去の $x$ の値だけに基づいた将来の $x$ の予測と、現在と過去の $x$ と $y$ の値に基づいた将来の $x$ の予測を比較して後者のMSEのほうが小さくなる場合、 $y_t$ から $x_t$ へのグレンジャー因果性 (Granger causality) が存在するといわれる。沖本 [2010]

メモリープールのデータについて解説を行う。次に線形回帰分析を行い価格と情報変数の関係について説明する。その後マルコフ転換モデルを活用し、線形回帰分析では不十分であった情報変数と価格変動の関係性についてさらに詳しく分析を行う。最後に本研究の成果と課題をまとめる。

## II 本論

### 1 データの説明

#### 1.1 トランザクション手数料

トランザクションコストの値を直接計算することは、トランザクションにかかるデータの量を計算することがビットコインの構造上不可能であることから、極めて難しい。

そこで本研究では、1つのブロックごとの取引手数料の総額の推移を観察する。ビットコインのトランザクションは1ブロック当たり1MBの制約があるため、この方法でもトランザクション手数料を分析することがある程度可能である。

トランザクション総額の推移は図5のようになり、その対数差分系列は図6のようになる。

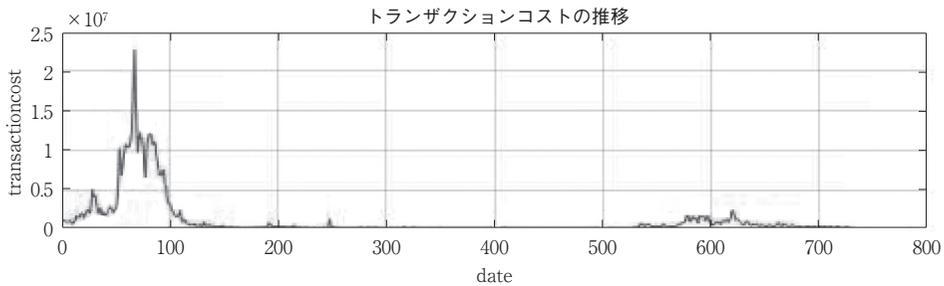


図5 トランザクションコストの時系列データ (2017/10/16~2019/10/15)

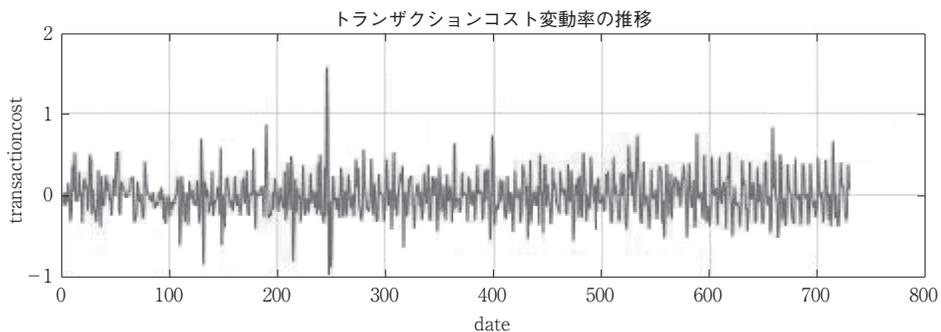


図6 トランザクション手数料の対数差分系列

#### 1.2 メモリープール

次にメモリープールのデータについて説明する。メモリープールのデータも Blockchain [2019] のデータを活用した。Blockchain [2019] ではメモリープールを、承認待ち状態のトランザクションの総数と定めている。このデータは、注1にも表記した通り、厳密にはデイリーデータではな

いが、このサイトのデータが最適であると判断したためこのデータを活用する。メモリープールの対数差分系列は図7のようになっている。

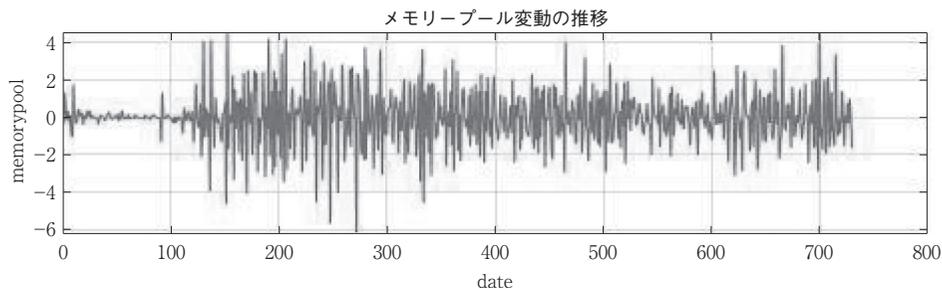


図7 メモリープールの対数差分系列

## 2 線形回帰分析を用いた分析

### 2.1 ビットコインと各変数の線形回帰分析

まず、ビットコインとそれぞれの時系列データに関する関係を理解するために、ビットコインを目的変数、トランザクション手数料、メモリープールのそれぞれを説明変数として、線形回帰分析を行った。線形回帰分析を行った結果以下のような結果が得られた。

表8 線形回帰分析の結果

データ	定数項	標準誤差	係数	標準誤差	決定係数
トランザクション手数料	6840.094015	90.37958861	0.000804782	0.000032265	0.467876
メモリープール	6703.455537	104.0834913	0.00005689	0.00000283	0.357530

### 2.2 分析の考察と線形回帰分析の限界

ビットコインと各変数の線形回帰分析を行い、表8のような結果が得られた。表8より、決定係数は十分には高くないものの、トランザクション手数料やメモリープールが大きい時のほうがビットコインの価格が大きい傾向にある。しかし、これらの結果が示していることは、ビットコインの決済機能の利点からすると、不可解な結果である。なぜなら、ビットコインの意義の所でも述べたように、決済機能としてのビットコインの優位性は、第三者を仲介しないことにより「手数料を低く抑えられる」ことにあるからである。そこで、私は以下のような仮説を考えた。

- ① ビットコインの価格の上昇は知名度の向上など別の要因にある。
- ② ビットコインの価格はトランザクションコストによって以下のように循環している
  - (ア) ビットコインの需要が大きくなるにつれて、ビットコイン決済需要が増加し、トランザクション手数料が増加、もしくはトランザクションしきれない取引が増加することによりメモリープールの量が増加。
  - (イ) トランザクション手数料やメモリープールの増加によって決済機能としてのビットコインの優位性がなくなりビットコインの価格が低下。
  - (ウ) 価格が下がったビットコインに対して送金需要が減少、トランザクション手数料、メモリー

プールの値が減少。

(エ) トランザクション手数料，メモリープールの値が減少したことにより，ビットコインの決済手段としての優位性が復活し価格が再び上昇する。

(オ) (イ) に戻る

この過程を検証するために必要なことは3つあると考える。

- ① ビットコインの上昇トレンドと下降トレンドをとらえる。
- ② それぞれの変数とトレンドの変化の関係を明らかにする。
- ③ 情報があつた時にどのトレンドにあるかを識別する。

そしてこの三つの条件に合った検証を行うために私は，Diebold, Lee, and Weinbach [1994] で紹介された方法で検証を行った。

### 3 本研究のモデルの紹介

#### 3.1 今回使用するマルコフ転換モデルの紹介

今研究のモデルにおいては，ビットコインの価格変動率が正規分布に従うという仮定を置く。図9は正規分布プロットといい Mathworks [2019] によると，正規分布に近い場合，プロットは赤い直線に沿うようになる。この図よりビットコインの価格変動率は正規分布と比較したときに極めて fat-tail であることがわかる。しかし，多くの先行研究においてビットコインを正規分布と仮定してマルコフ転換モデルを使用していることから，この仮定を置くことは妥当である。

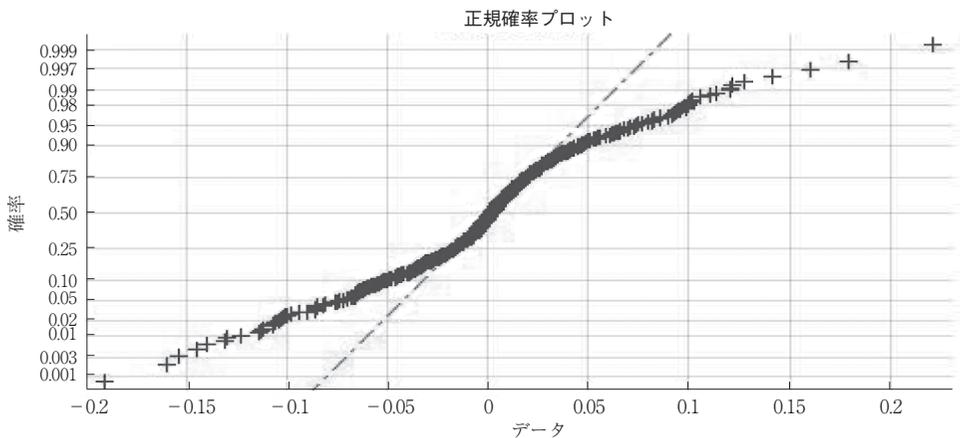


図9 ビットコイン価格変動率の正規確率プロット

ここからは，まず Kim and Nelson [1999] の4章に従い，一般的なマルコフ転換モデルについて説明を行い，その後に Diebold, Lee, and Weinbach [1994] の説明に基づき，本研究のマルコフ転換モデルについて解説する。

2つのレジームがあるモデルについて考える。

$$y_t = x_t \beta_{s_t} + e_t, \quad t=1,2,\dots,T, \quad e_t \sim N(0, \sigma_{s_t}^2) \quad (1.1)$$

$$\beta_{s_t} = \beta_0(1-S_t) + \beta_1 S_t \quad (1.2)$$

$$\sigma_{s_t}^2 = \sigma_0^2(1-S_t) + \sigma_1^2 S_t \quad (1.3)$$

$$S_t = 0 \text{ or } 1 \text{ (レジーム 0 か 1)} \quad (1.4)$$

これらの変数について説明する。 $y_t$  は目的変数で  $x_t$  は外生的なベクトル、 $\beta_{S_t}$  は  $x$  にかかるパラメーターであり  $\sigma_{S_t}^2$  は誤差項の分散である。

この式よりレジーム 0 の時は  $\beta_0$  と  $\sigma_0^2$  がパラメーターとして与えられ、レジーム 1 の時は  $\beta_1$  と  $\sigma_1^2$  がパラメーターとして与えられる。 $S_t$  が既知の場合はダミー変数モデルとなる。マルコフ転換モデルのパラメーターを推定する際には対数尤度関数を最大化する手法を利用する。対数尤度関数は

$$\ln L = \sum_{t=1}^T \ln(f(y_t | S_t)) \quad (1.5)$$

であり、 $f(y_t | S_t)$  は式 (1.6) によって与えられる

$$f(y_t | S_t) = \frac{1}{\sqrt{2\pi\sigma_{S_t}^2}} \exp\left(-\frac{\{y_t - x_t\beta_{S_t}\}^2}{2\sigma_{S_t}^2}\right) \quad (1.6)$$

この対数尤度関数におけるパラメーターは実際に観測することができないため、2つの段階によって値を決定する。

第一段階 まず  $y_t$  の同時分布と、 $S_t$  の値を考える。

$$f(y_t, S_t | \phi_{t-1}) = f(y_t | S_t, \phi_{t-1}) f(S_t | \phi_{t-1}) \quad (1.7)$$

なお、 $\phi_{t-1}$  は  $t-1$  期までの情報のことを示している。

第二段階 全ての  $S_t$  に対して  $f(y_t | S_t, \phi_{t-1})$  の期待値をとり  $f(S_t | \phi_{t-1})$  を得る。

$$\begin{aligned} f(y_t | \phi_{t-1}) &= \sum_{S_t=0}^1 f(y_t, S_t | \phi_{t-1}) \\ &= \sum_{S_t=0}^1 f(y_t | S_t, \phi_{t-1}) f(S_t | \phi_{t-1}) \\ &= \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left(-\frac{\{y_t - x_t\beta_0\}^2}{2\sigma_0^2}\right) \Pr[S_t=0 | \phi_{t-1}] + \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{\{y_t - x_t\beta_1\}^2}{2\sigma_1^2}\right) \Pr[S_t=1 | \phi_{t-1}] \end{aligned} \quad (1.8)$$

そして、対数尤度関数は

$$\ln L = \sum_{t=1}^T \ln\left\{\sum_{S_t=0}^1 f(y_t | S_t, \phi_{t-1}) \Pr[S_t | \phi_{t-1}]\right\} \quad (1.9)$$

で与えられる。この式の計算をするためには  $\Pr[S_t | \phi_{t-1}]$  がどれくらいの値であるかを計算することが必要不可欠である。今回のモデルではマルコフ転換モデルを使用する。 $r$  次のマルコフ転換モデルでは  $S_t$  の値は  $S_{t-1}, S_{t-2}, \dots, S_{t-r}$  までの過去の値に依存する。今回のモデルでは 1 次のマルコフ転換モデルについて扱うので  $S_t$  の値は  $S_{t-1}$  の値に依存する。

一般的なマルコフ転換モデルにおいて、状態変化確率は以下の式で計算される。

$$\Pr[S_t=1 | S_{t-1}=1] = p = \frac{\exp(p_0)}{1 + \exp(p_0)} \quad (1.10)$$

$$\Pr[S_t=0 | S_{t-1}=0] = q = \frac{\exp(q_0)}{1 + \exp(q_0)} \quad (1.11)$$

この計算を行う上では、各地点において状態  $S_t$  がどれくらいの確率でどのレジームにいるのか

を計算する必要がある。計算方法としては、その時点までの情報を活用して計算を行うフィルタ化と T 期までの全情報を利用して計算を行う平滑化の二つの手法がある。今回のモデルで使した、それぞれの手法の計算手順については、付録にて解説する。

次に今回のマルコフ転換モデルと通常マルコフ転換モデルの違いについて Diebold, Lee, and Weinbach [1994] の説明に基づいて解説を行う。

2.1.2 で述べたように、従来のモデルは状態変化確率に対して景気の変動などの経済ファンダメンタルズの要素を入れることが不可能である。一方でこのモデルでは、外部の情報を入れつつ、状態変化確率を内生変数として導出することができる。具体的には状態 1 と状態 2<sup>8)</sup> の二つのレジームが存在すると仮定し、 $\beta_1, \beta_2$  をパラメーターとしてロジスティック関数を作る。式にして表すと、

$$\Pr[S_t=1|S_{t-1}=1]=p_t^{11}=\frac{\exp(x_{t-1}\beta_1)}{1+\exp(x_{t-1}\beta_1)} \quad (1.12)$$

$$\Pr[S_t=2|S_{t-1}=1]=p_t^{12}=1-\frac{\exp(x_{t-1}\beta_1)}{1+\exp(x_{t-1}\beta_1)} \quad (1.13)$$

$$\Pr[S_t=1|S_{t-1}=2]=p_t^{21}=1-\frac{\exp(x_{t-1}\beta_2)}{1+\exp(x_{t-1}\beta_2)} \quad (1.14)$$

$$\Pr[S_t=2|S_{t-1}=2]=p_t^{22}=\frac{\exp(x_{t-1}\beta_2)}{1+\exp(x_{t-1}\beta_2)} \quad (1.15)$$

になる。これ以外の仮定については通常マルコフ転換モデルと同様のプロセスで計算を行い、フィルタ化と平滑化の方法で得られた確率変数をもとに計算を行う。

今回のモデルでは、状態変数  $S_t$  は 1 と 2 の二つの値をとる。情報変数  $x_t$  は  $2 \times T$  の行列であり、1 列目はすべて 1 が入り、2 列目にトランザクション手数料とメモリープールのそれぞれの変動率を代入する。 $\beta_1, \beta_2$  はそれぞれ  $\beta_1=(\beta_{10}, \beta_{11}), \beta_2=(\beta_{20}, \beta_{21})$  の  $1 \times 2$  のベクトルであり、 $\beta_{10}, \beta_{20}$  はそれぞれ定数項になり、 $\beta_{11}, \beta_{21}$  は情報変数にかかるパラメーターになる。今回の分析では、まず情報変数の影響を受けた二つのレジームがどのような特徴を持つか、そして、それぞれのレジームの転換に対して情報変数はどのような影響を及ぼしているかを分析する。具体的には計算によって導出された平均  $\mu$  が高いほうのレジームを上昇期、 $\mu$  の値が低いほうのレジームを下降期と定め、上昇期のレジーム、下降期のレジームへの状態変化確率に対して情報変数  $x_t$  が正の影響を及ぼすのか負の影響を及ぼすのかを  $\beta_{11}, \beta_{21}$  の推定値を計算することによって考察する。

これらの計算プロセスを実施し対数尤度関数を推定するために必要不可欠なのが EM アルゴリズムである。次節において Diebold, Lee, and Weinbach [1994] において利用されている EM アルゴリズムの計算プロセスについて解説を行う。

### 3.2 EM アルゴリズムについての説明

この説ではまず、Kim and Nelson [1999] の 4.3.5 節の EM アルゴリズムの内容にそって説明を行い、そのあと今回のマルコフ転換モデルにどのように適応されたかについて説明を行う。なお詳細な数式及び計算に使用したコードは付録に記載する。

8) Diebold, Lee, and Weinbach [1994] では状態 0, 状態 1 の二つのレジームとしているが、今回のモデルでは状態 1 と状態 2 の二つのレジームと定義したためこのように表記している。

EM アルゴリズムの手法は Dempster, Laird, and Rubin [1977] によって生み出された手法であり、不完全なデータもしくは不完全な値を含んだモデルの尤度関数を推定する代替的方法として生み出された。EM アルゴリズムは  $\theta$  を未知のパラメーターが入ったベクトルと定義すると、 $\theta$  の値の下で観測できない値の期待値を計算する “Expectation” のステップと計算された期待値をもとに  $\theta$  のパラメーターを最大化させる “Maximization” の二つのステップを繰り返すアルゴリズムである。具体的には

1.  $k-1$  回目の反復によって得られたパラメーターの推定値  $\theta^{k-1}$  を利用して、潜在変数の期待値を測定する。
2. 1. によって得られた潜在変数の条件付き期待値を利用して、パラメーターの尤度関数を最大化させ、 $\theta^k$  の値を計算する。

なおこの計算を行う際、 $\theta^0$  はこちらが設定を行い、またこのアルゴリズムは  $|\theta^k - \theta^{k-1}|$  が十分に小さくなるまで計算を行う。

次に EM アルゴリズムが今回のモデルにどのように適応されているかについて Diebold, Lee, and Weinbach [1994] の内容に基づいて説明を行う。

まず今回のマルコフ転換モデルにおいて  $\theta$  のパラメーターは以下の要素で構成される。

1. レジーム 1 及びレジーム 2 の平均 ( $\mu_1, \mu_2$ ) 及び分散 ( $\sigma_1^2, \sigma_2^2$ )
2. それぞれのレジームの状態変化確率関数内のパラメーター ( $\beta_{10}, \beta_{11}, \beta_{20}, \beta_{21}$ )  
 である。また、状態変化確率の計算のために第一期確率である  $\rho_1, \rho_2$  も初期値として与える。  
 これらを踏まえたうえで、今回使用する EM アルゴリズムのステップを説明すると以下のようになる。

1. 適当な初期値  $\theta^0$  を与える。
2. 与えられたパラメーターの状態確率及び平滑化した状態変化確率を求める。つまり、

$$P(S_t=1 | y_T, x_T; \theta^0) \quad \forall t \quad (2.1)$$

$$P(S_t=2 | y_T, x_T; \theta^0) \quad \forall t \quad (2.2)$$

$$P(S_t=1, S_{t-1}=1 | y_T, x_T; \theta^0) \quad \forall t \quad (2.3)$$

$$P(S_t=1, S_{t-1}=2 | y_T, x_T; \theta^0) \quad \forall t \quad (2.4)$$

$$P(S_t=2, S_{t-1}=1 | y_T, x_T; \theta^0) \quad \forall t \quad (2.5)$$

$$P(S_t=2, S_{t-1}=2 | y_T, x_T; \theta^0) \quad \forall t \quad (2.6)$$

また、これらの確率を使って  $E[\log(y_T, S_t | x_T; \theta^0)]$  を計算する。

3. 計算した平滑化した状態変化確率から  $\theta^1$  を計算する。

$$\theta^1 = \arg \max_{\theta} E[\log(y_T, S_t | x_T; \theta)] \quad (2.7)$$

4. 2. と 3. のステップを繰り返し、 $|\theta^1 - \theta^0|$  が収束するまで計算する。

なお今回のモデルでは、パラメーターの値の大きさも加味した結果収束判断としては、すべてのパラメーターの中で差が最大のものの差が  $10^{-7}$  よりも小さいことを条件とした。なお E ステップ

と M ステップの具体的な計算過程は巻末の付録に記載する。

## 4 結果と考察

### 4.1 それぞれの結果について

それぞれの情報変数について計算を行ったところ以下のような結果となった。なお、この時  $\pi_1\pi_2$  についてはそれぞれ、(1.12) における  $p_i^{11}$  と、(1.15) における  $p_i^{22}$  を意味している。

表 10 それぞれの結果一覧

トランザクション手数料								
パラメーター	$\mu_1$	$\mu_2$	$\sigma_1$	$\sigma_2$	$\beta_{10}$	$\beta_{11}$	$\beta_{20}$	$\beta_{21}$
結果	-0.000903	0.002060	0.003354	0.000258	2.621439	8.196010	2.020619	-4.332608
メモリープール								
パラメーター	$\mu_1$	$\mu_2$	$\sigma_1$	$\sigma_2$	$\beta_{10}$	$\beta_{11}$	$\beta_{20}$	$\beta_{21}$
結果	0.0003734	0.0006556	0.00310135	0.00086978	42.733199	21.759099	14.177958	-5.743039
データなし								
パラメーター	$\mu_1$	$\mu_2$	$\sigma_1$	$\sigma_2$	$\pi_1$	$\pi_2$		
結果	0.0020	-0.0007	0.0002	0.0032	0.8352	0.8622		

モデルの評価としてはより情報を多く利用して計算した平滑化確率を軸として評価を行った。

今回、情報変数をトランザクション手数料、メモリープール、もしくは情報変数なしの場合で計算した時すべての場合について高収益低分散のレジームと、低収益高分散のレジームに分かれた。

またトランザクションコストとメモリープールの時について、 $\beta$  の値は両方とも  $\beta_{11} > 0, \beta_{21} < 0$  の値を計測した。これが意味することとは、情報変数が正の値をとるとき（つまり、トランザクション手数料が増加もしくは、メモリープールの量が増加したとき）、レジーム 1 からはレジーム 1 にとどまりやすくなり一方でレジーム 2 にいるときはレジーム 1 に推移しやすくなる。一方で情報変数が負の値をとる（つまり、トランザクション手数料が減少及び、メモリープールの量が減少）場合、レジーム 1 からはレジーム 2 に移行する確率が上がる一方で、レジーム 2 からはレジーム 2 にとどまる確率が上昇する。つまりこの表を見る限りでは取引コストの上昇がビットコインの価格を下落するレジームに移行を促し、逆に取引コストの下落がビットコインの価格を上昇させることを促しているように見える。

次節ではこの考察が正しいかどうかをビットコインの実際の価格と比較して考察する。

## 4.2 各レジームとビットコインの変動に関する考察

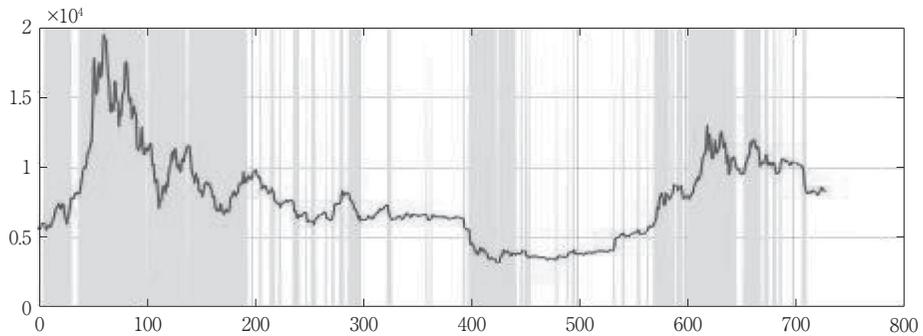


図 11 データなしの時のレジーム 2 の時とビットコインの価格の関係

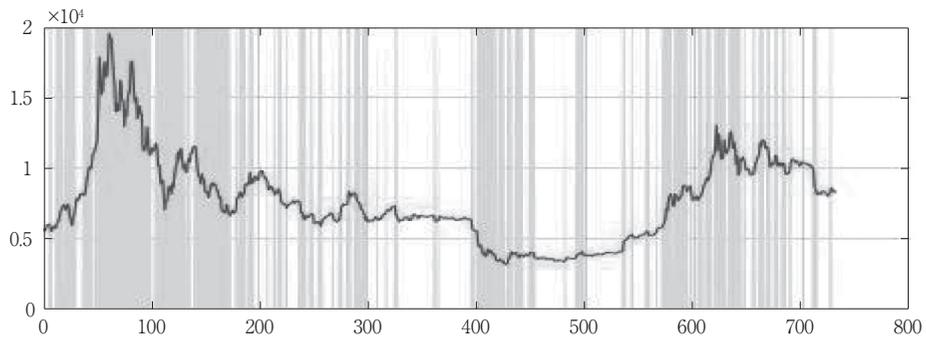


図 12 トランザクション手数料が情報変数におけるレジーム 1 とビットコインの価格の関係

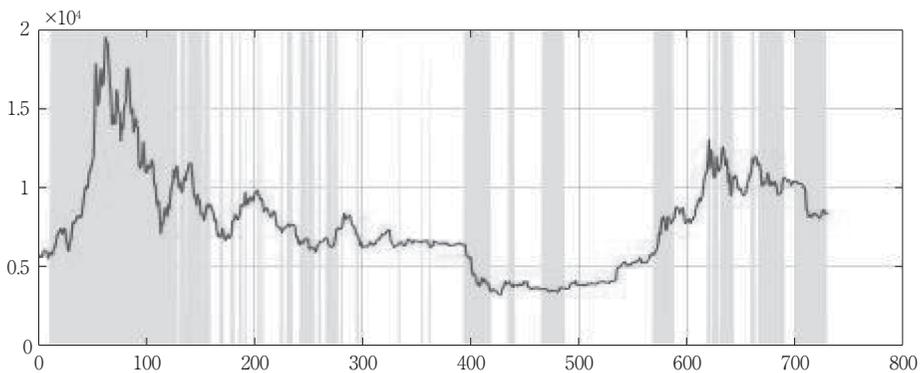


図 13 メモリープールが情報変数におけるレジーム 1 とビットコインの価格の関係

前節で、トランザクション手数料もしくはメモリープールの量の増加がビットコインの下降レジームに誘導するということが分かった。しかし、3つの方法すべてにおいて、2017年末にビットコイン価格が高騰しているときに対しても下降レジームにいると判断していることから、レジーム 1 とレジーム 2 をビットコイン価格の上昇、下降という分け方は適当ではなく、ビットコインの

変動率に注目し、変動期と安定期という分け方をした方が適切であるといえる。

また、トランザクションコストのデータを代入したものと通常のマルコフ転換モデルにおいて、図を見たら明らかであるようにレジームの評価としてはほとんど一致した結果となった。そのため、トランザクションコストの情報はビットコイン価格の変動に影響があるものの平滑化確率を用いたレジームの判定については、あまり影響を及ぼさないといえる。

またメモリープールに関しては、2017年末から2018年初頭にかけてほぼすべてレジーム1と判定した一方で2018年末の増減に関しては一部増減が激しいところもレジーム2という評価を行った。これはメモリープールのデータの変動率が2018年中旬以降著しく大きいことからこのような結果になったと推測される。

これらのレジーム判定から自分が線形回帰分析をした時の循環仮設は明確に否定されたわけではないものの、レジームと価格のグラフを見る限り、価格そのものよりもその変動の大きさが重要であることがいえることが分かった。

### Ⅲ 結論

#### 1 本研究の成果

今回のモデル検証では、 $\beta$ の値によってレジームの移行確率を計算することで情報変数の値がビットコインの価格変化に対してどのように影響を与えているかについて、ある程度検証することができた。

#### 2 今回のモデルの限界

今回のモデルにおいてはいくつかの限界点も見つかった。一つはこのモデルにおいて標準誤差を出すことができなかつたことである。理由としては本研究で参照した、Diebold, Lee and Weinbach [1994] においては標準誤差についての言及が一切ないことや、本研究で使用したモデルは一般的なマルコフ転換モデルと異なり、推移確率の値は一定ではなく、不確定な値があることから、解析的に標準誤差を出すことが極めて難しいことがあげられる。

また、このモデルではEMアルゴリズムを用いるが、EMアルゴリズムが導出する値は極値収束の可能性を排除することができない。

これらの点においては本研究の限界があったことは認めざるを得ない。

#### 3 今後の課題

この研究において今後の課題としては主に三点あげられる。まず一点目には予測精度の向上があげられる。今回の研究では、レジームの転換の回数が非常に多く、特にフィルター化確率においてはレジームの予想は困難な結果になっている。今後の研究では、レジームの転換がより少なくなり、将来のレジームの予想が容易になるようなモデルに改善したい。二点目は、ビットコイン以外の仮想通貨に関する研究である。現在では、イーサリアムやビットコインキャッシュなど多くの仮想通貨が作られている。それらの中にはマイニングの容量問題や、発行数を改善するなど、ビットコインネットワークの問題点を解決しているものもある。それらの通貨に関しても、今回と同様の検証を行った際に同じ結論が得られるのかは今後の課題である。三点目にはレジームの数の変更が

あげられる。今回のモデルでは、ビットコイン価格の増加減少の方向しか見ていなかったため、計算の容易な2つのレジームでの計算を行った。もしレジームの数を3つなどに増やしたときに同じ結論が得られるのかどうかは今後の課題である。

#### Ⅳ 謝辞

本研究では指導教官の江上教授に多くの助言をいただいた。また、ルースダン講師にはモデルのコーディングに関して貴重な助言をいただいた。さらに学部2回生だった時のゼミの担任の秋田教授にはEMアルゴリズムの計算に苦戦していた時に高性能のコンピューターを提供していただき、計算をスムーズに行うことができた。また、仮想通貨同好会の皆さんには、仮想通貨の仕組みについて議論していく中で仮想通貨に対してより深い理解をすることができた。これらすべての協力があって自分の修士論文は書き上げることができた。深く感謝したい。

#### 参考文献

- IT用語辞典バイナリ [2019]「クローリングとは何? Weblio 辞典」<https://www.weblio.jp/content/%E3%82%AF%E3%83%AD%E3%83%BC%E3%83%AA%E3%83%B3%E3%82%B0> (2019年12月1日アクセス)
- 沖本竜義 [2010]『経済・ファイナンスデータの計量時系列分析』朝倉書店
- 永井武彦 [2018]「キャッシュレス社会への取り組み」経済産業省 (<https://www.kantei.go.jp/jp/singi/keizaisai/sei/miraitoshikaigi/sankankyougikai/fintech/dail/siryou2.pdf>) 2019年11月24日アクセス
- 日下弘樹 [2019]「仮想通貨マイニングのブロック報酬の半減期とは? ライトコインの価格高騰との関連性やセキュリティにどのような影響を与えるのか」(<https://crypto.watch.impress.co.jp/docs/theme/1190780.html>) 2019年12月22日アクセス
- 二神孝一・堀敬一 [2009]『マクロ経済学 第二版』有斐閣 p100
- 日本経済新聞 [2014]「マウントゴックス破綻ビットコイン 114 億円消失」([https://www.nikkei.com/article/DGXNASGC2802\\_C\\_Y4A220C1MM8000/](https://www.nikkei.com/article/DGXNASGC2802_C_Y4A220C1MM8000/)) 2019年11月23日アクセス
- ルースダンケヴヘイッシュウィリ [2013]「マルコフ転換モデルによるバブル期の分析」
- Andreas M. Antonopoulos [2015] Mastering Bitcoin Unlocking Digital Cryptocurrency, O'Reilly (今井崇也, 鳩貝良一郎訳『ビットコインとブロックチェーン』NTT出版株式会社, 2016年)
- Atomic Wallet [2019] Bitcoin Scaling Problem, Explained (<https://atomicwallet.io/bitcoin-scaling-problem-explained>) 2019年11月30日アクセス
- Bitcoin Forum [2019] (<https://bitcointalk.org/>) 2019年12月28日アクセス
- Bitcoin Transaction Fees [不明] (<https://bitcoinfees.info/>) 2019年11月23日アクセス
- Blockchain [2019]“Memory pool size the aggregate size of transaction waiting to be confirmed” (<https://www.blockchain.com/ja/charts/mempool-size?timespan=all>) 2019年10月15日アクセス
- Captainaltcoin.com [2018] What is Bitcoin Memory Pool (<https://captainaltcoin.com/bitcoin-memory-pool-mempool/>) 2019年11月23日アクセス
- Daniel R. Chappell [2018] “Regime heteroscedasticity in Bitcoin: A comparison of Markov switching models,” MPRApaperNo. 90682
- David Ardia, Keven Bluteau, Maxime Rüede [2019] “Regime changes in Bitcoin GARCH volatility dynamics,” Finance Research Letters 29, pp. 266-271
- Dempster, Laird and Rubin [1977] “Maximum Likelihood from Incomplete Data Via the EM Algorithm,” Journal of

- the Royal Statistical Society: Series B (Methodological) 39, pp. 1-38
- Edwin Sin, Lipo Wang [2017] "Bitcoin Price Prediction Using Ensembles of Neural Networks, 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD 2017)"
- Francis X. Diebold, Joon-Haeng Lee and Gretchen C. Weinbach [1994] Regime switching with time-varying transition probabilities, Oxford:Oxford University Press, pp.283-302
- Hamilton, J. D. [1989] "A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle," *Econometrica* 57(2) pp. 357-384
- Jose A. Scheinkman and Wei Xiong [2003] "Overconfidence and Speculative Bubbles," *Journal of Political Economy* 111, pp. 1183-1219
- Kim, C. J. [1994] "Dynamic Linear Models with Markov-Switching," *Journal of Econometrics* 60(1-2), pp. 39-70.
- Kim C. J. and Nelson C. R. [1999] *State-Space Models with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications*, MIT Press
- MacLachlan and Krishnan [2008] *The EM Algorithm and Extensions*, John Wiley& Sons, Inc
- Martina Matta, Ilaria Leushe, Michele Marchesi [2015]
- XiangLi, ChongAlexWang [2017] "The technology and economic determinants of cryptocurrency exchange rates: The case of Bitcoin, *Decision Support Systems* 95, pp. 49-60"
- S. Nakamoto [2008] "Bitcoin: A Peer-to-Peer Electronic Cash System" *dhimmel/bitcoin-whitepaper@a5f36b3*
- TradeBlock [2019] "HistoricalData" (<https://tradeblock.com/bitcoin/historical/>) 2019年12月28日アクセス
- Young Bin Kim et al. [2016] "Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies," *Plos one* 11(8)

## 付録 1

### 1. Matlab のコード

この節では EM アルゴリズムのコードを記す。

関数コード `tvem.m`

```
function[new_output, filtprob, smoothedP_one, smoothedP_two, convergence_meter]=tvem(par,data,info)
T=length(data);
transp=zeros(2,2*(T-1));%transition probabilities which change on time
jcd=zeros(4,T-1);%joint conditional distribution of (y,s,t-1)
cl=zeros(1,T-1);%conditional likelihood
stateprob=zeros(4,T-1);%time-t filtered state probabilities
filtprob=zeros(2,T-1);%filtration probability
all=zeros(1,T-1);
new_miu=[par(1) par(2)];% 平均
new_variance=[par(3) par(4)];% 分散
new_beta1=[par(5) par(6)];
new_beta2=[par(7) par(8)];% 情報 X に対する係数
rho=[par(9) par(10)];% 第一期確率
f=zeros(2,T);
z=zeros(2,T-1);
convergence_meter=ones(1,8);
%for i = 1:100
while max(convergence_meter)>0.0000001
    disp(convergence_meter)
```

```

betal=new_betal;
beta2=new_beta2;
miu=new_miu;
variance=new_variance;
for t=1:T%Y の分布確率を 1 期から T 期まで計算(ステップ 1)
    f(1,t)=1/sqrt(2*pi*variance(1,1))*exp(-(data(t,1)-miu(1,1))^2/(2*variance(1,1)));
    f(2,t)=1/sqrt(2*pi*variance(1,2))*exp(-(data(t,1)-miu(1,2))^2/(2*variance(1,2)));
end
for t=1:T-1% 第一期から第二期より T-1 期から T 期までの transp(時間変化で変わる状態変化確率)を計算(ステップ 1)k=2
    transp(1,2*t)=exp(betal(1,1)+betal(1,2)*info(t,1))/(1+exp(betal(1,1)+betal(1,2)*info(t,1)));
    transp(2,2*t)=exp(beta2(1,1)+beta2(1,2)*info(t,1))/(1+exp(beta2(1,1)+beta2(1,2)*info(t,1)));
    transp(1,2*t)=1-transp(1,2*t-1);
    transp(2,2*t-1)=1-transp(2,2*t);
end
%transp
% 条件付同時確率を計算する (Calculate filtered joint state probabilities) (2-a~2-d)
% 条件付同時分布の二期の値を計算
jcd(1,1)=f(1,2)*transp(1,1)*rho(1,1);
jcd(2,1)=f(2,2)*transp(1,2)*rho(1,1);
jcd(3,1)=f(1,2)*transp(2,1)*rho(1,2);
jcd(4,1)=f(2,2)*transp(2,2)*rho(1,2);
cl(1,1)=sum(jcd(:,1));%y の条件付尤度を計算(2 期)
for k=1:4
    stateprob(k,1)=jcd(k,1)/cl(1,1);
    z(1,1)=f(1,2)/cl(1,1);
    z(1,2)=f(2,2)/cl(1,1);
end
for t=3:T%3 期以降の値について計算
    jcd(1,t-1)=f(1,t)*(transp(1,2*(t-1)-1))*(stateprob(1,t-2)+stateprob(3,t-2));
    jcd(2,t-1)=f(2,t)*(transp(1,2*(t-1)))*(stateprob(1,t-2)+stateprob(3,t-2));
    jcd(3,t-1)=f(1,t)*(transp(2,2*(t-1)-1))*(stateprob(2,t-2)+stateprob(4,t-2));
    jcd(4,t-1)=f(2,t)*(transp(2,2*(t-1)))*(stateprob(2,t-2)+stateprob(4,t-2));
    cl(1,t-1)=sum(jcd(:,t-1));
    z(1,t-1)=f(1,t)-cl(1,t-1);
    z(2,t-1)=f(2,t)-cl(1,t-1);
    for k=1:4
        stateprob(k,(t-1))=jcd(k,(t-1))/cl(1,(t-1));
    end
    filtprob(1,t-1)=stateprob(1,t-1)+stateprob(3,t-1);
    filtprob(2,t-1)=stateprob(2,t-1)+stateprob(4,t-1);
end
% 平滑化した同時状態確率を計算する(ステップ 3)
% 時間 1 から 2 の時が 1 列目になるようになるため一つ少ない
jointprob_oo_start=zeros(4,T-1);%t-1=1 t=1 の初期値(ステップ 3a)

```

```

jointprob_ot_start=zeros(4,T-1);%t-1=1 t=2
jointprob_to_start=zeros(4,T-1);%t-1=2 t=1
jointprob_tt_start=zeros(4,T-1);%t-1=2 t=2
smoothedJST_oneone=zeros(1,T);%smoothed joint state probability for time t(ステップ 3b)
smoothedJST_onetwo=zeros(1,T);
smoothedJST_twoone=zeros(1,T);
smoothedJST_twotwo=zeros(1,T);
smoothedP_one=zeros(1,T);%平滑化限界状態確率(ステップ 4)
smoothedP_two=zeros(1,T);
smoothedP_one(1,1)=rho(1,1);
smoothedP_two(1,1)=rho(1,2);
for t=3:T%3a の第一項についての計算(tau t-1 の 2^3 パターンしかない。該当しないパターンはゼロとして処理する)
    jointprob_oo_start(1,t-1)=(f(1,t)*transp(1,2*(t-1)-1)*stateprob(1,t-2))/cl(1,t-1);
    jointprob_oo_start(2,t-1)=(f(2,t)*transp(1,2*(t-1))*stateprob(1,t-2))/cl(1,t-1);
    jointprob_ot_start(3,t-1)=(f(1,t)*transp(2,2*(t-1)-1)*stateprob(2,t-2))/cl(1,t-1);
    jointprob_ot_start(4,t-1)=(f(2,t)*transp(2,2*(t-1))*stateprob(2,t-2))/cl(1,t-1);
    jointprob_to_start(1,t-1)=(f(1,t)*transp(1,2*(t-1)-1)*stateprob(3,t-2))/cl(1,t-1);
    jointprob_to_start(2,t-1)=(f(2,t)*transp(1,2*(t-1))*stateprob(3,t-2))/cl(1,t-1);
    jointprob_tt_start(3,t-1)=(f(1,t)*transp(2,2*(t-1)-1)*stateprob(4,t-2))/cl(1,t-1);
    jointprob_tt_start(4,t-1)=(f(2,t)*transp(2,2*(t-1))*stateprob(4,t-2))/cl(1,t-1);
end
for t=2:T-2%2 期から T-2 期について計算
    jointprob_oneone=zeros(4,T);
    jointprob_onetwo=zeros(4,T);
    jointprob_twoone=zeros(4,T);
    jointprob_twotwo=zeros(4,T);
    jointprob_oneone(:,t)=jointprob_oo_start(:,t);
    jointprob_onetwo(:,t)=jointprob_ot_start(:,t);
    jointprob_twoone(:,t)=jointprob_to_start(:,t);
    jointprob_twotwo(:,t)=jointprob_tt_start(:,t);
    for tau=t+2:T
        jointprob_oneone(1,tau-1)=f(1,tau)*transp(1,2*(tau-1)-1)*(jointprob_oneone(1,tau-2)+jointprob_oneone(3,tau-2))/cl(1,tau-1);
        jointprob_oneone(2,tau-1)=f(2,tau)*transp(1,2*(tau-1))*(jointprob_oneone(1,tau-2)+jointprob_oneone(3,tau-2))/cl(1,tau-1);
        jointprob_oneone(3,tau-1)=f(1,tau)*transp(2,2*(tau-1)-1)*(jointprob_oneone(2,tau-2)+jointprob_oneone(4,tau-2))/cl(1,tau-1);
        jointprob_oneone(4,tau-1)=f(2,tau)*transp(2,2*(tau-1))*(jointprob_oneone(2,tau-2)+jointprob_oneone(4,tau-2))/cl(1,tau-1);
        jointprob_onetwo(1,tau-1)=f(1,tau)*transp(1,2*(tau-1)-1)*(jointprob_onetwo(1,tau-2)+jointprob_onetwo(3,tau-2))/cl(1,tau-1);
        jointprob_onetwo(2,tau-1)=f(2,tau)*transp(1,2*(tau-1))*(jointprob_onetwo(1,tau-2)+jointprob_onetwo(3,tau-2))/cl(1,tau-1);
        jointprob_onetwo(3,tau-1)=f(1,tau)*transp(2,2*(tau-1)-1)*(jointprob_onetwo(2,tau-2)+jointprob_onetwo(4,tau-2))/cl(1,tau-1);
        jointprob_onetwo(4,tau-1)=f(2,tau)*transp(2,2*(tau-1))*(jointprob_onetwo(2,tau-2)+jointprob_onetwo(4,tau-2))/cl(1,tau-1);
    end
end

```

```

tprob_onetwo(4,tau-2))/cl(1,tau-1);
    jointprob_onetwo(4,tau-1)=f(2,tau)*transp(2,2*(tau-1))*(jointprob_onetwo(2,tau-2)+ jointprob_
onetwo(4,tau-2))/cl(1,tau-1);
    jointprob_twoone(1,tau-1)=f(1,tau)*transp(1,2*(tau-1)-1)*(jointprob_twoone(1,tau-2)+ join-
tprob_twoone(3,tau-2))/cl(1,tau-1);
    jointprob_twoone(2,tau-1)=f(2,tau)*transp(1,2*(tau-1))*(jointprob_twoone(1,tau-2)+ jointprob_
twoone(3,tau-2))/cl(1,tau-1);
    jointprob_twoone(3,tau-1)=f(1,tau)*transp(2,2*(tau-1)-1)*(jointprob_twoone(2,tau-2)+ join-
tprob_twoone(4,tau-2))/cl(1,tau-1);
    jointprob_twoone(4,tau-1)=f(2,tau)*transp(2,2*(tau-1))*(jointprob_twoone(2,tau-2)+ jointprob_
twoone(4,tau-2))/cl(1,tau-1);
    jointprob_twtwo(1,tau-1)=f(1,tau)*transp(1,2*(tau-1)-1)*(jointprob_twtwo(1,tau-2)+ join-
tprob_twtwo(3,tau-2))/cl(1,tau-1);
    jointprob_twtwo(2,tau-1)=f(2,tau)*transp(1,2*(tau-1))*(jointprob_twtwo(1,tau-2)+ join-
tprob_twtwo(3,tau-2))/cl(1,tau-1);
    jointprob_twtwo(3,tau-1)=f(1,tau)*transp(2,2*(tau-1)-1)*(jointprob_twtwo(2,tau-2)+ join-
tprob_twtwo(4,tau-2))/cl(1,tau-1);
    jointprob_twtwo(4,tau-1)=f(2,tau)*transp(2,2*(tau-1))*(jointprob_twtwo(2,tau-2)+ join-
tprob_twtwo(4,tau-2))/cl(1,tau-1);
end

smoothedJST_oneone(1,t)=sum(jointprob_oneone(:,T-1));
smoothedJST_onetwo(1,t)=sum(jointprob_onetwo(:,T-1));
smoothedJST_twoone(1,t)=sum(jointprob_twoone(:,T-1));
smoothedJST_twtwo(1,t)=sum(jointprob_twtwo(:,T-1));

smoothedP_one(1,t)=(smoothedJST_oneone(1,t)+ smoothedJST_twoone(1,t));
smoothedP_two(1,t)=(smoothedJST_onetwo(1,t)+ smoothedJST_twtwo(1,t));
end
%1 期についての計算

smoothedP_one(1,1)=smoothedJST_oneone(1,2)+ smoothedJST_onetwo(1,2);

smoothedP_two(1,1)=smoothedJST_twoone(1,2)+ smoothedJST_twtwo(1,2);
%T-1 期についての計算
    smoothedJST_oneone(1,T-1)=sum(jointprob_oo_start(:,T-1));
    %smoothedJST_onetwo(1,T-1)=sum(jointprob_ot_start(:,T-1));
    %smoothedJST_twoone(1,T-1)=sum(jointprob_to_start(:,T-1));
    smoothedJST_twtwo(1,T-1)=sum(jointprob_tt_start(:,T-1));
    smoothedP_one(1,T-1)=jointprob_oo_start(1,T-1)+ jointprob_oo_start(2,T-1)+ jointprob_to_start(1,
T-1)+ jointprob_to_start(2,T-1);
    smoothedP_two(1,T-1)=jointprob_ot_start(3,T-1)+ jointprob_ot_start(4,T-1)+ jointprob_tt_start(3,
T-1)+ jointprob_tt_start(4,T-1);
%T 期についての計算(フィルタリングの流用)
    smoothedJST_oneone(1,T)=stateprob(1,T-1);

```

```

%smoothedJST_onetwo(1,T)=stateprob(2,T-1);
%smoothedJST_twoone(1,T)=stateprob(3,T-1);
smoothedJST_twotwo(1,T)=stateprob(4,T-1);
smoothedP_one(1,T)=filtprob(1,T-1);
smoothedP_two(1,T)=filtprob(2,T-1);
%mean(jointprob_tt_start) + mean(jointprob_oo_start) + mean(jointprob_ot_start) + mean(jointprob_to_
start)
% 平滑化確率の合計値の確認に使用。誤差 1.2%が望ましい。
%M_algorithm,solve FOC and derive miu,variance and rho value
miu_vector=zeros(T,2);
var_vector=zeros(T,2);
for t=1:T
    miu_vector(t,1)=data(t,1)*smoothedP_one(1,t);
    miu_vector(t,2)=data(t,1)*smoothedP_two(1,t);
end
new_miu(1,1)=sum(miu_vector(:,1))/sum(smoothedP_one(1,:));
new_miu(1,2)=sum(miu_vector(:,2))/sum(smoothedP_two(1,:));
for t=1:T
    var_vector(t,1)=(data(t,1)-miu(1,1))^2*smoothedP_one(1,t);
    var_vector(t,2)=(data(t,1)-miu(1,2))^2*smoothedP_two(1,t);
end
new_variance(1,1)=sum(var_vector(:,1))/sum(smoothedP_one(1,:));
new_variance(1,2)=sum(var_vector(:,2))/sum(smoothedP_two(1,:));
rho(1,1)=smoothedP_one(1,1);
rho(1,2)=smoothedP_two(1,1);
%M_algorithm,solve FOC and derive beta (b0 and b1 is have two value for each t)
%make vectors for calculating beta
beta_dash10=zeros(T-1,1);%beta の微分
beta_dash11=zeros(T-1,1);
beta_dash20=zeros(T-1,1);
beta_dash21=zeros(T-1,1);
beta_bunnshiu=zeros(T-1,2);%beta の分子のそれぞれの項の計算
beta_bunnshid=zeros(T-1,2);
beta_bunnboa=zeros(T-1,2);
beta_bunnbob=zeros(T-1,2);
beta_bunnboc=zeros(T-1,2);
beta_bunnbod=zeros(T-1,2);
beta_bunnbo_matrix1=zeros(2,2);%beta の分母のそれぞれの項の計算
beta_bunnshi_matrix1=zeros(2,1);
beta_bunnbo_matrix2=zeros(2,2);
beta_bunnshi_matrix2=zeros(2,1);
for t= 2:T%k=2 の式を活用
    beta_dash10(t-1,1)=(exp(beta(1,1) + beta(1,2)*info(t-1,1))/(1 + exp(beta(1,1) + beta(1,2)*info
(t-1,1))) - (exp(beta(1,1) + beta(1,2))/(1 + exp(beta(1,1) + beta(1,2)*info(t-1,1))))^2);
    beta_dash11(t-1,1)=info(t-1,1) * (exp(beta(1,1) + beta(1,2)*info(t-1,1))/(1 + exp(beta(1,1) +

```

```

beta1(1,2)*info(t-1,1))- (exp(beta1(1,1)+ beta1(1,2))/(1 + exp(beta1(1,1)+ beta1(1,2)*info(t-1,1))))^2);
    beta_dash20(t-1,1)=(exp(beta2(1,1)+ beta2(1,2)*info(t-1,1))/(1 + exp(beta2(1,1)+ beta2(1,2)*info
(t-1,1)))- (exp(beta2(1,1)+ beta2(1,2))/(1 + exp(beta2(1,1)+ beta2(1,2)*info(t-1,1))))^2);
    beta_dash21(t-1,1)=info(t-1,1)*(exp(beta2(1,1)+ beta2(1,2)*info(t-1,1))/(1 + exp(beta2(1,1)+
beta2(1,2)*info(t-1,1)))- (exp(beta2(1,1)+ beta2(1,2))/(1 + exp(beta2(1,1)+ beta2(1,2)*info(t-1,1))))^2);
    beta_bunboa(t-1,1)=smoothedP_one(1,t-1)*beta_dash10(t-1,1);
    beta_bunbob(t-1,1)=smoothedP_one(1,t-1)*beta_dash11(t-1,1);
    beta_bunboc(t-1,1)=info(t-1,1)*smoothedP_one(1,t-1)*beta_dash10(t-1,1);
    beta_bunbod(t-1,1)=info(t-1,1)*smoothedP_one(1,t-1)*beta_dash11(t-1,1);
    beta_bunshiu(t-1,1)=(smoothedJST_oneone(1,t)-smoothedP_one(1,t-1)*(exp(beta1(1,1)+ info(t-1,
1)*beta1(1,2))/(1 + exp(beta1(1,1)+ info(t-1,1)*beta1(1,2))))-beta_dash10(t-1,1)*beta1(1,1)-beta_dash11
(t-1,1)*beta1(1,2));
    beta_bunshid(t-1,1)=info(t-1,1)*(smoothedJST_oneone(1,t)-smoothedP_one(1,t-1)*(exp(beta1(1,
1)+ info(t-1,1)*beta1(1,2))/(1 + exp(beta1(1,1)+ info(t-1,1)*beta1(1,2))))-beta_dash10(t-1,1)*beta1(1,1)
-beta_dash11(t-1,1)*beta1(1,2));
    beta_bunboa(t-1,2)=smoothedP_two(1,t-1)*beta_dash20(t-1,1);
    beta_bunbob(t-1,2)=smoothedP_two(1,t-1)*beta_dash21(t-1,1);
    beta_bunboc(t-1,2)=info(t-1,1)*smoothedP_two(1,t-1)*beta_dash20(t-1,1);
    beta_bunbod(t-1,2)=info(t-1,1)*smoothedP_two(1,t-1)*beta_dash21(t-1,1);
    beta_bunshiu(t-1,2)=(smoothedJST_twotwo(1,t)-smoothedP_two(1,t-1)*(exp(beta2(1,1)+ info
(t-1,1)*beta2(1,2))/(1 + exp(beta2(1,1)+ info(t-1,1)*beta2(1,2))))-beta_dash20(t-1,1)*beta2(1,1)-beta_
dash21(t-1,1)*beta2(1,2));
    beta_bunshid(t-1,2)=info(t-1,1)*(smoothedJST_twotwo(1,t)-smoothedP_two(1,t-1)*(exp(beta2(1,
1)+ info(t-1,1)*beta2(1,2))/(1 + exp(beta2(1,1)+ info(t-1,1)*beta2(1,2))))-beta_dash20(t-1,1)*beta2(1,1)
-beta_dash21(t-1,1)*beta2(1,2));
    end
    %300 ページの式計算
    beta_bunbo_matrix1(1,1)=sum(beta_bunboa(:,1));
    beta_bunbo_matrix1(1,2)=sum(beta_bunbob(:,1));
    beta_bunbo_matrix1(2,1)=sum(beta_bunboc(:,1));
    beta_bunbo_matrix1(2,2)=sum(beta_bunbod(:,1));
    beta_bunbo_matrix2(1,1)=sum(beta_bunboa(:,2));
    beta_bunbo_matrix2(1,2)=sum(beta_bunbob(:,2));
    beta_bunbo_matrix2(2,1)=sum(beta_bunboc(:,2));
    beta_bunbo_matrix2(2,2)=sum(beta_bunbod(:,2));
    beta_bunshi_matrix1(1,1)=sum(beta_bunshiu(:,1));
    beta_bunshi_matrix1(2,1)=sum(beta_bunshid(:,1));
    beta_bunshi_matrix2(1,1)=sum(beta_bunshiu(:,2));
    beta_bunshi_matrix2(2,1)=sum(beta_bunshid(:,2));

new_beta1=transpose(beta_bunbo_matrix1¥beta_bunshi_matrix1);

new_beta2=transpose(beta_bunbo_matrix2¥beta_bunshi_matrix2);

new_output=[new_miu(1,1),new_miu(1,2),new_variance(1,1),new_variance(1,2),new_beta1(1,1),new_beta1

```

```

(1,2),new_beta2(1,1),new_beta2(1,2)];

output=[miu(1,1),miu(1,2),variance(1,1),variance(1,2),beta1(1,1),beta1(1,2),beta2(1,1),beta2(1,2)];
convergence_meter=abs(new_output-output);

end

スクリプトファイル scripttvem.m
bit_price=importdata('C:\Users\cism\Desktop\ronbun_data\bitcoin_twoyear-2.csv');
%trans_cost=importdata('C:\Users\cism\Desktop\ronbun_data\transaction-fees_twoyear.csv');
%mempool=importdata('C:\Users\cism\Desktop\ronbun_data\mempool_daily_twoyear_2.csv');
data=zeros(729,1);
info=zeros(729,1);
for t =1:729
data(t,1)=log(bit_price(t + 1,1)/bit_price(t,1));
%info(t,1)=log(trans_cost.data(t + 1,1)/trans_cost.data(t,1));
%info(t,1)=log(mempool(t + 1,1)/mempool(t,1));
end
beta1=1
beta2=-1
p1=exp(beta1)/(1 + exp(beta1));
p2=exp(beta2)/(1 + exp(beta2));
m=sum(data)/729;
%initial=[0.0003734;0.0006556;0.00310135;0.00086978;42.733199;21.759099;14.177958;-5.74304;0;1];
%initial=[-0.000903;0.00206;0.003354;0.000258;2.621439;8.196010;2.020619;-4.3326790;1;0];
%initial2=[0.002011797900075;-0.000728682611612;0.000234708051842;0.003235393573295;0.835151638262381;
0.862162453994054]
initial2=[0.001;-0.001;0.0017;0.0017;p1;p2;0.5;0.5];
format long
%[a,b,c,d,e]=tvem(initial,data,info);
[a,b,c,d,e]=emem(initial2,data);

```

## 付録 2

### 2 EM アルゴリズムの計算過程

この節では実際にモデルに使用した EM アルゴリズムの計算過程について解説する。途中でフィルター化確率と平滑化確率についての計算を行うのでそこでフィルター化確率と平滑化確率のそれぞれの解説も行う。なお、この計算過程はすべて Francis X. Diebold, Joon-Haeng and Grechen C Weinbach [1994] の説明に従っている。

#### 2.1 EM アルゴリズムの E ステップにおける計算過程

まず、E ステップにおいては、観測されたデータを利用して尤度関数の期待値を計算する。E ステップの計算は以下の順序で行う。

##### 2.1.1 ステップ 1

それぞれの  $t$  時点における、 $y_t$  の条件付分布と、状態変化確率を計算する。

$$(y_t | S_t = i; \alpha_i) \stackrel{iid}{\sim} N(\mu_i, \sigma_i^2), \quad (1)$$

$$f(y_t | S_t = i; \alpha_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{y_t - \mu_i^2}{2\sigma_i^2}\right) \quad (2)$$

$$p_i^{00} = \frac{\exp x_{t-1} \beta_0}{1 + \exp x_{t-1} \beta_0} \quad (3)$$

$$p_i^{01} = 1 - \frac{\exp x_{t-1} \beta_0}{1 + \exp x_{t-1} \beta_0} \quad (4)$$

$$p_i^{10} = 1 - \frac{\exp x_{t-1} \beta_1}{1 + \exp x_{t-1} \beta_1} \quad (5)$$

$$p_i^{11} = \frac{\exp x_{t-1} \beta_1}{1 + \exp x_{t-1} \beta_1} \quad (6)$$

## 2.1.2 ステップ 2

フィルター化確率を以下の手順で計算する。フィルター化確率はその時点より前の情報だけを用いて計算する確率である。

### 2.1.2.1 2A 同時条件付分布の計算

同時条件付分布を計算で求める。 $\underline{y}_{t-1}, \underline{x}_{t-1}$  の情報に基づいて、 $t=2$  の時は以下のように計算する。

$$f(y_2, s_2, s_1 | y_1, x_1; \theta^{(j-1)}) = f(y_2 | s_2; \alpha^{(j-1)}) P(s_2 | s_1, x_1; \beta^{(j-1)}) P(s_1) \quad (7)$$

$t=3$  以降については以下のように計算する。

$$f(y_t, s_t, s_{t-1} | \underline{y}_{t-1}, \underline{x}_{t-1}; \theta^{(j-1)}) = \sum_{s_{t-2}=0}^1 f(y_t | s_t, s_{t-1} | \underline{y}_{t-1}, \underline{x}_{t-1}; \beta^{(j-1)}) P(s_{t-1}, s_{t-2} | \underline{y}_{t-1}, \underline{x}_{t-1}; \theta^{(j-1)}) \quad (8)$$

なおこの時の条件付分布と状態変化確率はステップ 1 のを活用する。

### 2.1.2.2 2B 条件付尤度関数の計算

条件付尤度関数  $y_t$  をそれぞれの状態の同時条件付分布を合計して計算する。

$$f(y_t | \underline{y}_{t-1}, \underline{x}_{t-1}; \theta^{(j-1)}) = \sum_{s_{t-1}=0}^1 \sum_{s_{t-2}=0}^1 f(y_t, s_t, s_{t-1} | \underline{y}_{t-1}, \underline{x}_{t-1}; \theta^{(j-1)}) \quad (9)$$

### 2.1.2.3 2C フィルター化確率の計算

$t$  時点のフィルター化された状態確率を計算する。分子に 2A で計算した同時条件付分布、分母に 2B で計算した条件付尤度関数を代入する。

$$P(s_t, s_{t-1} | \underline{y}_t, \underline{x}_t; \theta^{(j-1)}) = \frac{f(y_t, s_t, s_{t-1} | \underline{y}_t, \underline{x}_t; \theta^{(j-1)})}{f(y_t | \underline{y}_t, \underline{x}_t; \theta^{(j-1)})} \quad (10)$$

### 2.1.2.4 2D 繰り返し

2a から 2c のステップを T-2 回繰り返す。

## 2.1.3 ステップ 3

今度は平滑化した状態変化確率を計算する。平滑化確率は満期までの情報すべてを使い確率の値を計算する。

### 2.1.3.1 3A 同時確率の計算

まず  $t=2$  であるとする。この時所与である  $(s_t, s_{t-1})$  と  $\underline{y}_t, \underline{x}_t$  から同時確率  $(s_\tau, s_{\tau-1}, s_t, s_{t-1})$  を計算する。まず  $\tau=t-1$  の時はその計算は以下の式から導出される。

$$P(s_{t+1}, s_t, s_{t-1} | \underline{y}_{t+1}, \underline{x}_{t+1}; \theta^{(j-1)}) = \frac{f(y_{t+1} | s_{t+1}; \alpha^{(j-1)}) P(s_{t+1} | s_t, x_t; \beta^{(j-1)}) P(s_t, s_{t-1} | \underline{y}_t, \underline{x}_t; \theta^{(j-1)})}{f(y_{t+1} | \underline{y}_{t+1}, \underline{x}_{t+1}; \theta^{(j-1)})} \quad (11)$$

次に  $\tau=t+2$  以降の時は以下のように導出する。

$$\begin{aligned} & P(s_\tau, s_{\tau-1}, s_t, s_{t-1} | \underline{y}_\tau, \underline{x}_\tau; \theta^{(j-1)}) \\ &= \frac{\sum_{s_{\tau-2}=0}^1 f(y_\tau | s_\tau; \alpha^{(j-1)}) P(s_\tau | s_{\tau-1}, x_{\tau-1}; \beta^{(j-1)}) P(s_{\tau-1}, s_{\tau-2}, s_t, s_{t-1} | \underline{y}_{\tau-1}, \underline{x}_{\tau-1}; \theta^{(j-1)})}{f(y_\tau | \underline{y}_{\tau-1}, \underline{x}_{\tau-1}; \theta^{(j-1)})} \end{aligned} \quad (12)$$

それぞれの  $\tau$  について計算を行い、 $\tau=T$  になるまで計算を繰り返す。

### 2.1.3.2 3B 平滑化同時状態確率の計算

$\tau=T$  にまで到達したら T 時点と T-1 時点のそれぞれの状態確率について足し合わせ、 $(s_t, s_{t-1})$  の確率を導出す

る。

$$P(s_t, s_{t-1} | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) = \sum_{s_T=0}^1 \sum_{s_{T-1}=0}^1 P(s_T, s_{T-1}, s_t, s_{t-1} | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) \quad (13)$$

### 2.1.3.3 3C 繰り返し計算

ステップ3aとステップ3bを繰り返し、それぞれの状態についての  $(s_t, s_{t-1})$  の値を計算する

### 2.1.3.4 3D 繰り返し計算

$t=3, 4, \dots, T$  についても同様に計算する。

(注意)  $t=T-1, t=T$  のときに同様に計算することはできない。よって  $t=T-1$  の時は式(11)を使用して計算を行い  $t=T$  の時はフィルター化確率と同じ値が出るのでフィルター化確率を使用して計算した。これは Francis X. Diebold, Joon-Haeng and Gretchen C Weinbach [1994] では見落とされている。

### 2.1.4 ステップ4 平滑化した周辺状態確率の計算

平滑化した周辺状態確率は平滑化した状態変化確率の合計として求められる。以下の式は一つの例である。

$$P(s_t=1 | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) = P(s_t=1, s_{t-1}=0 | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) + P(s_t=1, s_{t-1}=1 | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) \quad (14)$$

## 2.2 EM アルゴリズムの M ステップ

次に M ステップの解説をする。平滑化状態変化確率が所与である中で全データを使った対数尤度関数は、モデルパラメータの制約において直接的に最大化されている。これらの尤度関数の一階条件を計算することでパラメータの値を計算することができる。

### 2.2.1 $\mu, \sigma, \rho$ の第一期確率の計算

$\mu, \sigma, \rho$  は以下の一階条件を解くことで求められる。

$$\sum_{i=1}^T P(s_i | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) (y_i - \mu_i^{(j)}) = 0 \quad (15)$$

$$\sum_{i=1}^T P(s_i | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) \left( \frac{y_i - \mu_i^{(j)}}{(\sigma_i^{(j)})^2} - 1 \right) = 0 \quad (16)$$

$$P(s_1 | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) \left( \frac{1}{\rho} \right) - 1 = 0 \quad (17)$$

よってこれらの式を解いてやると

$$\mu_i^{(j)} = \frac{\sum_{t=1}^T P(s_t | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) (y_t)}{\sum_{t=1}^T P(s_t | \underline{y}_T, \underline{x}_T; \theta^{(j-1)})} \quad (18)$$

$$(\sigma_i^{(j)})^2 = \frac{\sum_{t=1}^T P(s_t | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) (y_t - \mu_i^{(j)})^2}{\sum_{t=1}^T P(s_t | \underline{y}_T, \underline{x}_T; \theta^{(j-1)})} \quad (19)$$

$$\rho^{(j)} = P(s_1=1 | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) \quad (20)$$

$$i=0, 1$$

### 2.2.2 $\beta$ の導出

次に  $\beta$  の導出過程を説明する。 $\beta$  の一階条件は以下の式で与えられる。

$$\sum_{i=2}^T x_{i-1} \{ P(s_i=0, s_{i-1}=0 | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) - p_i^{00} P(s_{i-1}=0 | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) \} = 0 \quad (21)$$

$$\sum_{i=2}^T x_{i-1} \{ P(s_i=1, s_{i-1}=1 | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) - p_i^{11} P(s_{i-1}=1 | \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) \} = 0 \quad (22)$$

この時  $p_i^{00}$  と  $p_i^{11}$  はテイラー展開の一時近似を利用する。つまり、

$$p_i^{00}(\beta_0) \approx p_i^{00}(\beta_0^{(j-1)}) + \frac{\partial p_i^{00}(\beta_0^{(j-1)})}{\partial \beta_0} \Big|_{\beta_0=\beta_0^{(j-1)}} (\beta_0 - \beta_0^{(j-1)}) \quad (23)$$

$$p_i^{11}(\beta_1) \approx p_i^{11}(\beta_1^{(j-1)}) + \frac{\partial p_i^{11}(\beta_1^{(j-1)})}{\partial \beta_1} \Big|_{\beta_1=\beta_1^{(j-1)}} (\beta_1 - \beta_1^{(j-1)}) \quad (24)$$

と近似をする。議論を単純化するために以下のように表記する。

$$p_{it}^{00}(\beta_0) = \frac{\partial p_t^{00}(\beta_0^{(j-1)})}{\partial \beta_{i0}} \Big|_{\beta_0 = \beta_0^{(j-1)}}, i=0,1 \quad (25)$$

$$p_{it}^{11}(\beta_1) = \frac{\partial p_t^{11}(\beta_1^{(j-1)})}{\partial \beta_{i1}} \Big|_{\beta_1 = \beta_1^{(j-1)}}, i=0,1 \quad (26)$$

よって確率のベクトルは以下のように与えられる。

$$\frac{\partial p_t^{00}(\beta_0^{(j-1)})}{\partial \beta_{i0}} \Big|_{\beta_0 = \beta_0^{(j-1)}} = (p_{it}^{00}(\beta_0^{(j-1)}), p_{it}^{00}(\beta_0^{(j-1)})) \quad (27)$$

$$\frac{\partial p_t^{11}(\beta_1^{(j-1)})}{\partial \beta_{i1}} \Big|_{\beta_1 = \beta_1^{(j-1)}} = (p_{it}^{11}(\beta_1^{(j-1)}), p_{it}^{11}(\beta_1^{(j-1)})) \quad (28)$$

でありそれぞれの要素は以下の式で導出できる。

$$p_{it}^{00}(\beta_0^{(j-1)}) = x_{i,t} - 1(p_{it}^{00}(\beta_0^{(j-1)}) - (p_{it}^{00}(\beta_0^{(j-1)}))^2), i=0,1 \quad (29)$$

$$p_{it}^{11}(\beta_1^{(j-1)}) = x_{i,t} - 1(p_{it}^{11}(\beta_1^{(j-1)}) - (p_{it}^{11}(\beta_1^{(j-1)}))^2), i=0,1 \quad (30)$$

これらの式を代入すると一階条件は以下のように変形する。

$$\sum_{t=2}^T x_{t-1} \left\{ P(s_t=0, s_{t-1}=0 \mid \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) - P(s_{t-1}=0 \mid \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) \left[ p_t^{00}(\beta_0^{(j-1)}) + \frac{\partial p_t^{00}(\beta_0^{(j-1)})}{\partial \beta_{i0}} \Big|_{\beta_0 = \beta_0^{(j-1)}} (\beta_0 - \beta_0^{(j-1)}) \right] \right\} = 0 \quad (31)$$

$$\sum_{t=2}^T x_{t-1} \left\{ P(s_t=1, s_{t-1}=1 \mid \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) - P(s_{t-1}=1 \mid \underline{y}_T, \underline{x}_T; \theta^{(j-1)}) \left[ p_t^{00}(\beta_0^{(j-1)}) + \frac{\partial p_t^{00}(\beta_0^{(j-1)})}{\partial \beta_{i0}} \Big|_{\beta_0 = \beta_0^{(j-1)}} (\beta_0 - \beta_0^{(j-1)}) \right] \right\} = 0 \quad (32)$$

これを  $\beta$  について解くと以下の式で表される。

$$\beta_0^{(j)} = \begin{pmatrix} \beta_{00}^{(j)} \\ \beta_{01}^{(j)} \end{pmatrix} = \begin{pmatrix} \sum_{t=2}^T x_{0,t-1} P(S_{t-1}=0) p_{0t}^{00} & \sum_{t=2}^T x_{0,t-1} P(S_{t-1}=0) p_{1t}^{00} \\ \sum_{t=2}^T x_{1,t-1} P(S_{t-1}=0) p_{0t}^{00} & \sum_{t=2}^T x_{1,t-1} P(S_{t-1}=0) p_{1t}^{00} \end{pmatrix}^{-1} \times \begin{pmatrix} \sum_{t=2}^T x_{0,t-1} \left\{ P(s_t=0, s_{t-1}=0) - P(s_{t-1}=0) \left[ p_t^{00} - \frac{\partial p_t^{00}}{\partial \beta_0} \beta_0^{(j-1)} \right] \right\} \\ \sum_{t=2}^T x_{1,t-1} \left\{ P(s_t=0, s_{t-1}=0) - P(s_{t-1}=0) \left[ p_t^{00} - \frac{\partial p_t^{00}}{\partial \beta_0} \beta_0^{(j-1)} \right] \right\} \end{pmatrix} \quad (33)$$

$$\beta_1^{(j)} = \begin{pmatrix} \beta_{10}^{(j)} \\ \beta_{11}^{(j)} \end{pmatrix} = \begin{pmatrix} \sum_{t=2}^T x_{0,t-1} P(S_{t-1}=1) p_{0t}^{11} & \sum_{t=2}^T x_{0,t-1} P(S_{t-1}=1) p_{1t}^{11} \\ \sum_{t=2}^T x_{1,t-1} P(S_{t-1}=1) p_{0t}^{11} & \sum_{t=2}^T x_{1,t-1} P(S_{t-1}=1) p_{1t}^{11} \end{pmatrix}^{-1} \times \begin{pmatrix} \sum_{t=2}^T x_{0,t-1} \left\{ P(s_t=1, s_{t-1}=1) - P(s_{t-1}=1) \left[ p_t^{11} - \frac{\partial p_t^{11}}{\partial \beta_1} \beta_1^{(j-1)} \right] \right\} \\ \sum_{t=2}^T x_{1,t-1} \left\{ P(s_t=1, s_{t-1}=1) - P(s_{t-1}=1) \left[ p_t^{11} - \frac{\partial p_t^{11}}{\partial \beta_1} \beta_1^{(j-1)} \right] \right\} \end{pmatrix} \quad (34)$$