

ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

DİNAMİK MODELLERDE MCMC YÖNTEMLERİ ÜZERİNE BİR ÇALIŞMA

Bahar AKTEPE

İSTATİSTİK ANABİLİM DALI

Ankara
2025

Her hakkı saklıdır

ÖZET

Yüksek Lisans Tezi

DİNAMİK MODELLERDE MCMC YÖNTEMLERİ ÜZERİNE BİR ÇALIŞMA

Bahar AKTEPE

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
İstatistik Anabilim Dalı

Danışman: Doç. Dr. Esin KÖKSAL BABACAN

Zaman içinde değişen sistemleri modelleyen yapılara dinamik modeller denir. Bayesci istatistik, bu tür modellerde belirsizlikleri ifade etmek ve geçmiş bilgiyle güncel veriyi birleştirerek çıkarım yapmak için kullanılan güçlü bir yaklaşımdır. Markov Zinciri Monte Carlo (MCMC) yöntemleri ise, bu çıkarım sürecinde karmaşık sonsal dağılımlardan örnekleme yaparak Bayesci analizin uygulanmasını mümkün kılan hesaplama araçlarıdır. Bu çalışmada MCMC yöntemlerinin kuramsal temelleri, başlıca algoritmaları (Metropolis-Hastings, Gibbs örnekleme, Hamiltonian Monte Carlo) ve birbirleriyle farkları detaylı biçimde ele alınmıştır. Ayrıca, bu yöntemlerin dinamik modeller üzerindeki uygulanabilirliğini göstermek amacıyla, Bitcoin log getiri serisi üzerinde açıklayıcı değişkenli bir stokastik volatilité modeli kurulmuş ve model, Hamiltonian Monte Carlo algoritması ile tahmin edilmiştir. Çalışma, Bayesci istatistik ve MCMC yöntemlerinin teorik yapısını açıklamakla kalmayıp, bu yöntemlerin pratikteki gücünü de görünür kılmayı amaçlamaktadır.

Haziran 2025, 56 sayfa

Anahtar Kelimeler: Dinamik Modeller, Bayesçi İstatistik, Markov Zinciri Monte Carlo, Hamiltonian Monte Carlo, Stokastik Volatilité

ABSTRACT

Master Thesis

A STUDY ABOUT MCMC METHODS IN DYNAMIC MODELS

Bahar AKTEPE

Ankara University
Graduate School of Natural and Applied Science
Department of Statistics

Supervisor: Asoc. Prof. Dr. Esin KÖKSAL BABACAN

Dynamic models are structures that represent systems evolving over time, typically involving unobserved latent states that generate the observed data. Bayesian statistics is a powerful framework for making inferences in such models by expressing uncertainty and combining prior information with current data. Markov Chain Monte Carlo (MCMC) methods are computational tools that enable Bayesian analysis by sampling from complex posterior distributions. This study presents the theoretical foundations of MCMC methods, their main algorithms (Metropolis-Hastings, Gibbs sampling, Hamiltonian Monte Carlo), and a detailed comparison of their properties. Furthermore, to demonstrate the applicability of these methods to dynamic models, a stochastic volatility model with explanatory variables is constructed using Bitcoin log return data and estimated via the Hamiltonian Monte Carlo algorithm. This thesis aims not only to explain the theoretical structure of Bayesian statistics and MCMC methods but also to reveal their practical power through a real-world application.

June 2025, 56 pages

Keywords: Dynamic Models, Bayesian Statistics, Markov Chain Monte Carlo, Hamiltonian Monte Carlo, Stochastic Volatility

ÖNSÖZ VE TEŞEKKÜR

Yüksek lisans tez çalışmalarım sırasında bilgi birikimini aktaran, tecrübesiyle yön veren ve desteklerini hiçbir zaman esirgemeyen saygıdeğer hocam sayın Doç. Dr. Esin KÖKSAL BABACAN'a (Ankara Üniversitesi İstatistik Bölümü Bilgi Sistemleri Anabilim Dalı) teşekkürlerimi sunarım.

Aldığım her kararda yanımda olan, beni bu günlere getiren, yol gösteren babam Hüseyin AKTEPE, annem Nihal AKTEPE, abim Umut AKTEPE ve eşi Zahide AKTEPE'ye; var oluş sebebim olan aileme sonsuz şükran ve teşekkürlerimi sunarım.

Son olarak tez çalışmamda desteklerini hep hissettiğim arkadaşlarım Aslı KARTAL, Berna BAYAR ve çalışma arkadaşlarım Mehmet DEMİRTAŞ ve Cenk TAŞDEMİR'e sonsuz teşekkürlerimi sunarım.

Bahar AKTEPE

Ankara, Haziran 2025

İÇİNDEKİLER

TEZ ONAY SAYFASI

ETİK.....	i
ÖZET.....	ii
ABSTRACT	iii
ÖNSÖZ VE TEŞEKKÜR.....	iv
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	vii
1. GİRİŞ ve ÖNCEKİ ÇALIŞMALAR.....	1
2. BAYESÇİ SONUÇ ÇIKARIMI.....	8
3. DAĞILIMLARDAN RASGELE SAYI ÜRETME	14
3.1 Ters Dönüşüm Yöntemi.....	14
3.2 Kabul-Red Yöntemi	15
4. MARKOV ZİNCİRİ VE MONTE CARLO YÖNTEMLERİ	18
4.1 Markov Zinciri	18
4.2 Monte Carlo İntegrasyonu	19
4.3 MCMC Yöntemlerinin Genel İşleyişi.....	20
4.3.1 Metropolis ve Metropolis-Hastings algoritması yöntemi	21
4.3.2 Gibbs örnekleme yöntemi	24
4.3.2.1 Sonsal dağılım ve Gibbs örnekleme	27
4.3.3 Hamiltonian Monte Carlo yöntemi.....	29
5. MCMC YÖNTEMİ İLE DAĞILIMLARDAN SAYI ÜRETME UYGULAMALARI	33
5.1 Cauchy Dağılımından Sayı Üretmek İçin Metropolis Algoritması	33
5.2 Normal Dağılım Parametrelerinin Sonsal Dağılımlarından Gibbs Örnekleme Yöntemi İle Örneklem Alınması	35
6. DİNAMİK MODELLERDE MCMC YÖNTEMLERİ ÜZERİNE GERÇEK VERİ UYGULAMASI	40
6.1 Modelin Yapısı.....	40
6.2 HMC Algoritması ile Günlük Bitcoin Volatilitésinin Modellenmesi.....	43
7. SONUÇ.....	50
7.1 Gelecek Çalışmalar İçin Öneriler	51
KAYNAKLAR	53
ÖZGEÇMİŞ.....	56

ŞEKİLLER DİZİNİ

Şekil 5.1 Metropolis Algoritması ile Cauchy dağılımından örnekleme.....	34
Şekil 5.2 Gibbs Algoritması ile Normal Dağılımın μ parametresinin sonsal dağılımında örnekleme	37
Şekil 5.3 Gibbs Algoritması ile Normal Dağılımın σ^2 parametresinin sonsal dağılımından örnekleme	37
Şekil 5.4 Normal Dağılımın μ parametresinin sonsal dağılımından elde edilen iz grafiği	38
Şekil 5.5 Normal Dağılımın σ^2 parametresinin sonsal dağılımından elde edilen iz grafiği	39
Şekil 6.1 Dört HMC zinciri için iz grafikleri (traceplot)	46
Şekil 6.2 Gerçekleşen log-getiri mutlak değerleri ($ y_t $) ve tahmin edilen volatilité ($\hat{\sigma}_t$).....	47
Şekil 6.3 10.04.2025-01.06.2025 arası tahmin edilen volatilité ve gerçekleşen getiriler	49

ÇİZELGELER DİZİNİ

Çizelge 1.1 Zaman, sıcaklık, ölçülen sıcaklık değerleri	3
Çizelge 6.1 Gelman-Rubin R değerleri anlam tablosu.....	45
Çizelge 6.2 $\hat{R}$ (Gelman-Rubin $\hat{R}$) ve n_{eff} değerleri	45



1. GİRİŞ ve ÖNCEKİ ÇALIŞMALAR

Markov Zinciri Monte Carlo (MCMC) yöntemi Monte Carlo simülasyonu ve Markov Zincirinin birleşiminden doğan, karmaşık dağılımlardan örnekleme yapmayı amaçlayan istatistiksel bir tekniktir. Bu yöntem matematiksel ve fiziksel problemlerin çözümüne yardımcı olur. Her iki kavramın kökeni eskiye dayanmaktadır, ancak MCMC'nin modern hali 20. Yüzyılın ortalarında geliştirilmiştir.

Markov Zinciri, Rus matematikçi Andrey Markov tarafından 1906 yılında tanıtılmıştır. Andrey Markov bir sonraki durumun sadece mevcut durumlara bağlı olduğu sistemleri incelemiştir (Markov, 1906; Seneta, 2006). Örneğin bir şehirde hava durumu bugün yağmurluysa, yarın nasıl olacağı sadece bugünkü duruma bağlı olabilir. Markov bu gibi durumları bir matematiksel model olarak formüle etmiştir (Hayes, 2023).

Monte Carlo simülasyonu, rastgele örneklerle karmaşık problemlerin çözülmesini sağlayan bir yöntemdir (Metropolis ve Ulam, 1949). Bu yöntem II. Dünya Savaşı sırasında içerisinde Stanislaw Ulam ve John von Neumann'ın da bulunduğu Amerikalı bir grup fizikçi tarafından nükleer reaksiyonları modellemek için geliştirilmiştir. Bu yöntemle çok sayıda rastgele deney gerçekleştirilerek, ilgilenilen niceliğin ortalaması ya da dağılımı gibi istatistiksel özellikleri yaklaşık olarak hesaplanır (Kroese, 2014). Ulam bir gün hasta yatağında solitaire oynarken oyunu kazanma olasılığının nasıl hesaplanabileceğini düşünmüş ve deterministik hesaplamalar zor olduğundan, rastgele deneyler yaparak bir ortalama bulmanın daha kolay olabileceğini fark etmiştir. Bu fikir Monte Carlo simülasyonunun temelini oluşturmuştur.

MCMC'nin modern şekli 1953 yılında Nicholas Metropolis ve meslektaşları tarafından geliştirilen Metropolis algoritması ile ortaya çıkmıştır. Ekip, nükleer reaksiyonlar ile ilgili hesaplamalar yaparken Markov Zinciri ile Monte Carlo yöntemini birleştirmiştir (Metropolis ve Ulam, 1949). Amaç karmaşık bir dağılımdan örnekler üreterek dağılımın istatistiksel özelliklerini tahmin etmektir. Daha sonra, 1970 yılında Hastings, Metropolis algoritması üzerinden Metropolis-Hastings algoritmasını geliştirmiştir (Hastings, 1970).

Bu algoritma, geniş bir dağılım yelpazesi için çalışabilen daha esnek bir versiyondur. İlerleyen süreçte, bilgisayarların da gücü arttıkça MCMC yöntemleri popülerleşmiş ve daha karmaşık-katmanlı hale gelmiştir. Bunlardan Gibbs algoritması, Stuart Geman ve Donald Geman tarafından 1984 yılında geliştirilmiş, Metropolis-Hastings' ten daha özel bir algoritma olup, büyük veri setlerinde daha verimli çalışabilmektedir. Hamiltonian Monte Carlo (HMC); fizik alanındaki Hamilton mekaniğinden esinlenilerek 1987 yılında geliştirilmiştir. Bu yöntem büyük veri kümelerinde daha az adımda etkili örnekleme yapabilen bir yöntemdir. Tempered MCMC ve Paralel MCMC ile 2000'li yıllarda birden fazla Markov Zinciri çalıştırılarak çok modlu dağılımlarda geçişi kolaylaştıran dağılımlar geliştirilmiştir. Bugün MCMC yöntemleri yapay zeka, makina öğrenimi, istatistik, genetik ve fizik, ekonomi gibi farklı bilim alanlarında kullanılmaktadır.

MCMC yöntemleri, statik (zamana bağlı olmayan) sistemlerden ve dinamik (zamana bağlı) sistemlerden örneklem oluşturmak için kullanılır. MCMC bağımlı örnekler üretir, ancak dinamik modellerde bu bağımlılık daha da belirgin hale gelir. Statik MCMC yöntemleri veriyi tek seferde işler. Dinamik modellerde her yeni örnek sadece bir önceki örnekten değil, aynı zamanda birden fazla önceki örnekten de etkilenebilir. Mesela durum uzay modelleri gibi sistemlerde gözlenen veri bir dizi gizli durumdan türetilir ve bu durumlar birbirine bağlıdır. Bu dinamik modele basit bir örnek aşağıdaki şekilde verilebilir.

Bir odanın sıcaklığını ölçen hassas olmayan bir termometre ele alınsın. Gerçek sıcaklık x_t ile ifade edilsin. Gerçek sıcaklık bilinmiyor, termometre ile ölçülüyor ve termometre ölçüm hataları yapıyor. Ancak her an sıcaklık ölçümü, buna y_t denilsin, yapılıyor. Burada amaç gerçek sıcaklık olan x_t 'yi tahmin etmek. Bu modelde dış ortam sıcaklığı değiştiğinde, mesela odada ocak yakıldığında rastgele bir değişiklik olacaktır, bu belirsizliğe w_t denilsin. Buraya kadar anlatılan model aşağıdaki biçimde ifade edebilir:

$$x_t = x_{t-1} + w_t . \quad (1.1)$$

Termometrenin ölçtüğü sıcaklık gerçek sıcaklığa yakındır ama ölçüm hataları içerir. Bu da şöyle gösterilebilir:

$$y_t = x_t + v_t \quad (1.2)$$

(v_t : ölçüm hatası, örneğin ± 2 °C olabilir)

Örneğin, gerçek sıcaklık başlangıçta $x_0 = 20$ °C olsun. Oda sıcaklığı her saniye rastgele biraz çıkıp azalsın, mesela $w_t \sim N(0,1)$ olsun. Termometrenin ölçüm hatası da $v_t \sim N(0,2)$ (± 2 °C civarında dalgalanıyor) olsun.

Çizelge 1.1 Zaman, sıcaklık, ölçülen sıcaklık değerleri

Zaman	x_t (bilinmiyor)	y_t
t=1	20,5 °C	21,2°C
t=2	19,8 °C	18,9 °C
t=3	21,0 °C	22,3 °C
t=4	20,2 °C	19,5 °C

Burada termometre verisinde her yeni ölçüm bir sonraki ölçümü etkilemekte ve veri seti bağımlı değişkenlerden oluşmaktadır. Bu tarz modellere dinamik modeller denir. Bu modelde MCMC yöntemleri kullanarak gerçek sıcaklık tahmin edilebilir.

Dinamik modeller, birçok bilimsel alanda ve mühendislik alanında sistemlerin zaman içindeki değişimini anlamak için kullanılır. Örneğin, finans piyasalarında hisse senedi fiyat tahminleri, hava durumu tahminleri, robot hareketleri, biyolojik sistemlerde salgın hastalık modellemeleri gibi birçok alanda kullanılır.

Termometre örneği bir de şu şekilde ele alınsın:

Bilinmeyen, tahmin edilmek istenilen gerçek sıcaklık x olsun. Elimizde termometreden herhangi bir anda alınan $(y_1, y_2, \dots, y_N)$ gözlemleri olsun ve bu ölçümlerin hatalar içerdiği bilinsin. Model şu şekilde ifade edilebilir:

$$y_t = x + v_t \quad (1.3)$$

($v_t \sim N(0, \sigma^2)$ ölçüm hatası)

Bu modelde gerçek sıcaklığı tahmin etmek için ölçülen y_t sıcaklık değerleri herhangi bir anda alınmış ve birbirinden bağımsız ölçümlerdir. Bu tarz modellere statik modeller denir. En çok olabilirlik yöntemi ile gerçek sıcaklık değeri tahmin edilebilir.

Statik modeller hızlı ve basittir. Ancak zamanla değişen sistemler için uygun değildir. Termometre örneğinde ölçümlerin herhangi bir anda yapıldığı düşünölsün. Eğer sıcaklık zamanla değişiyorsa, mesela odada pencere açıldı veya kalorifer yakıldı, birbirinden bağımsız yapılan ölçümler tüm süreci açıklamayacak, doğru şekilde temsil edemeyecektir. Sıcaklık ölçümleri eski ölçümleri güncelleyerek gitmeyeceği için sonuçta yapılacak tahmin yanıltıcı olacaktır.

Göröldüğü gibi, statik modeller zamanla değişen sistemlerde yetersiz kalmaktadır. Çünkü bu tür sistemlerde her yeni gözlem, sistemin mevcut durumunu güncelleyerek daha isabetli tahminler yapmayı gerektirir. Bu noktada dinamik süreçler zamana bağılı güncelleme yapılmasına izin vererek geçmişle bağı korur. Dinamik Markov Zinciri modellerde Monte Carlo (MCMC) yöntemleri kullanarak sadece geçmiş verilere bağı kalınmaz, aynı zamanda yeni gözlemler dikkate alınarak sistemin zaman içindeki değişimi daha doğru şekilde modellenir. Özellikle Kalman Filtreleri ve Parçacık MCMC gibi yöntemler, dinamik sistemlerin gizli durumlarını güncelleyerek en iyi tahminin yapılmasına olanak tanır. Bu da MCMC'yi zamana bağılı süreçleri analiz etmek için güçlü bir araç haline getirir.

İstatistikte MCMC'nin en yaygın kullanım alanlarından biri Bayesci istatistiktir. Bayesci istatistik, belirsizlik altında çıkarım yapmanın güçlü bir yoludur ve genellikle karmaşık olasılık dağılımlarından örnekleme yapmayı gerektirir. Ancak bu dağılımları doğrudan hesaplamak çoğu zaman zor veya imkansızdır. İşte bu noktada MCMC yöntemleri devreye girer. MCMC, Bayes teoremini kullanarak olasılıksal modellerin güncellenmesini sağlar ve özellikle yüksek boyutlu, karmaşık sistemlerde Bayesci

çıkarmı mümkün kılar. Dinamik sistemlerde ise gözlemlenen veriler ve gizli durumlar arasındaki bağımlılıkları takip ederek, zaman içinde değişen süreçleri modellemeye yardımcı olur.

Markov Zinciri Monte Carlo (MCMC) yöntemleri, modern bilim ve teknolojiye karmaşık olasılık yapılarının modellenmesini mümkün kılarak birçok alanda devrim yaratmıştır;

Gelfand ve Smith (1990), Gibbs örneklemesini istatistiksel çıkarıma entegre ederek her parametre için koşullu dağılımlardan sırasıyla örnekleme yapıp tüm modelin sonsal dağılımını yaklaşık olarak elde etmeyi sağlamıştır. Böylece Bayesci modellemenin pratik uygulanabilirliğini ortaya koymuş ve modern Bayesci istatistiğin temelini atmıştır.

Sims ve Zha (1998), çok değişkenli ekonomik modellerde MCMC tabanlı Bayesci yaklaşımı kullanarak para politikası şoklarının zaman içindeki etkilerini belirsizlikle birlikte değerlendirmiştir. Para politikası şoklarının zaman içindeki etkilerini belirsizlik aralıklarıyla değerlendirmek, ekonomi politikalarının sadece ortalama etkiyi değil, olası riskleri ve senaryoları da göstermesini sağlamıştır. Böylece bu alanda daha temkinli, veri temelli ve risk yönetimini içeren biçimde politika üretilmesi gerçekleşmiştir.

Rannala ve Yang (1996), DNA dizilim verilerinden türler arasındaki evrimsel ilişkileri (genetikte evrimsel ağaç yapılarını) çıkarırken Bayesci çerçevede tanımladıkları olasılık modeline dayanarak, ağaç yapıları ve model parametreleri için sonsal dağılımlardan MCMC ile örnekleme yapmıştır. Böylece sadece tek bir ‘en iyi ağaç’ yerine, veriye uygun birçok olası ağaç yapısını olasılıkla tartarak değerlendirebilmiştir. Bu yaklaşım, yeni türlerin sınıflandırılmasında, virüs mutasyonlarının izlenmesinde ve genetik soy analizlerinde daha esnek, veriye uyumlu ve bilimsel açıdan sağlam sonuçlar elde edilmesini mümkün kılmıştır.

Christensen ve arkadaşları (2001), evrenin parametrelerini (örneğin Hubble sabiti, madde yoğunluğu, eğrilik vs.) tahmin etmek için Bayesci çerçevede bir model kurmuştur. Bu modelin sonsal dağılımlarından, MCMC (özellikle Metropolis-Hastings) algoritmasını

kullanarak örnekleme yapmıştır. Böylece her parametre için yalnızca noktasal bir tahmin değil, tam bir olasılık dağılımı ve güven aralıkları elde etmiştir. Bu çalışma, kozmolojik parametre tahminlerinde deterministik yaklaşımlar yerine olasılıksal (Bayesci) çıkarımın kullanılabileceğini göstererek fiziksel kozmolojide paradigmayı değiştirmiştir. Artık evrenin yaşı, genişleme hızı gibi büyüklükler tek bir değerden ibaret değil; belirsizlikleriyle birlikte modellenen bilimsel büyüklükler haline gelmiştir. Bu yaklaşım, NASA'nın Planck ve WMAP gibi görevlerinde de benimsenerek, modern evren modellerinin temellerini oluşturmuştur ve MCMC'yi kozmolojide vazgeçilmez bir araç yapmıştır (Christensen vd., 2001; Planck Collaboration, 2018; WMAP Science Team, 2011).

Kastner ve Frühwirth-Schnatter (2014), ekonometrik ve finansal modelleme çalışmalarında kullanılan stokastik volatilité modelleri için sıkça karşılaşılan karışım sorununu çözmek amacıyla ASIS algoritması adlı örnekleme stratejisini geliştirmiştir. Bu strateji, model parametreleri arasındaki yüksek korelasyon nedeniyle MCMC zincirinin yavaş karışması ve etkili örnek sayısının düşük olması gibi sorunlara çözüm önerisi getirmiş, böylece MCMC'nin verimliliği artarken, özellikle finansal zaman serilerinde kullanılan stokastik volatilité modellerinde yüksek boyutlu ve karmaşık yapıların Bayesci olarak güvenilir biçimde tahmin edilmesi mümkün hale gelmiştir.

Bronk Ramsey (2009), arkeolojik karbon-14 tarihleme verilerini Bayesci MCMC yöntemleriyle analiz ederek hem kalibrasyon hem de birden fazla örnek üzerinden zamansal faz modelleri kurmuştur. Bu sayede, sadece tek bir tahmin değil; örnekler arasındaki belirsizlikler, sıralamalar ve evrensel takvimlere göre güven aralıkları da aynı anda elde edilmiş olur.

Griffiths ve arkadaşları (2008), insan zihninin karar verme ve kavramsal öğrenme süreçlerini Bayesci modellerle açıklamıştır. MCMC, bu modellerin parametrelerini ve yapısını hem teorik hem uygulamalı olarak keşfetmeyi sağlamış, zihinsel model öğrenimini simüle etmek için kullanılmıştır.

Bu alıřmalar gstermektedir ki MCMC yntemleri yalnızca matematiksel bir ara deęil, aynı zamanda istatistiksel dřnce biimini kkten deęiřtiren ve farklı bilim alanlarında keřfi tetikleyen ileri dzey bir yntemdir.

Bu alıřmada ilk olarak Bayesci istatistiksel yaklařımın temel kavramları ele alınacak, zellikle belirsizlik altında ıkarım yapma sreci ve nsel–sonsal daęılım iliřkisi aıklanacaktır. Ardından, rastgele sayı retme teknikleri incelenerek bilgisayar destekli simlasyon yntemlerinin temel dayanakları ortaya konulacaktır. Markov zincirlerinin teorik yapısı ve zamanla denge daęılımına yakınsama zellikleri aıklanacak; bunu takiben Monte Carlo yntemlerinin temel prensipleri ve yksek boyutlu problemlerde yaklařık hesaplamadaki rol anlatılacaktır. Tezin MCMC Yntemleri blmnde, Metropolis-Hastings, Gibbs rneklemesi, Hamiltonian Monte Carlo ve Parallel Tempering gibi temel MCMC algoritmaları ayrıntılı biimde ele alınacaktır. Ayrıca MCMC algoritmalarıyla daęılımlardan sayı retme sreci, Metropolis algoritması ile Cauchy daęılımı ve Gibbs rneklemesi ile Normal daęılım parametreleri zerinden uygulamalı rneklerle gsterilmiřtir.

Son olarak teorik olarak aıklanan MCMC algoritmalarının iřleyiřini gstermek amacıyla Bitcoin volatilitesinin stokastik modeli zerinde Hamiltonian Monte Carlo yntemi ile srec ve sonular gsterilmiř, yntemin aıklayıcılığı tartıřılmıřtır.

2. BAYESCI SONUÇ ÇIKARIMI

Bayesci istatistik, olasılığın bir olay hakkındaki mevcut bilgilere (ön bilgi-prior) dayalı olarak güncellenmesine olanak tanıyan bir yaklaşımdır (Gelman vd. 2013). Olasılığı yalnızca frekanslarla değil, belirsizlik karşısında ön bilgiye dayalı bir güven düzeyi olarak da ele alır. Frekansçı istatistikte olasılık, belirli bir olayın uzun vadeli gerçekleşme oranı olarak tanımlanırken, Bayesci istatistikte mevcut bilgiler ve yeni veriler doğrultusunda güncellenebilen bir inanç ölçüsü olarak yorumlanmaktadır.

Bayesci yaklaşım, 1763'te Thomas Bayes tarafından ortaya konulan ve matematiksel istatistiğin önemli bir teoremi olan Bayes teoremine dayanır. Bayes teoremi, 18. yüzyılda olasılık kuramının henüz gelişmekte olduğu bir dönemde, etkiden nedene ulaşmayı amaçlayan özgün bir düşünce deneyi ile ortaya konmuştur. Günümüzde ters olasılık (inverse probability) olarak bilinen bu yaklaşım, gözlemlenen sonuçlardan hareketle gizli bir nedenin olasılığını tahmin etmeye yöneliktir. Sharon Bertsch, McGrayne'nin "The Theory That Would Not Die" kitabında Bayes'in bu deneyi şöyle tasarladığını anlatmıştır; düz ve kare biçiminde bir masa göz önüne alınır. Bu masa o kadar düzgündür ki, üzerine atılan bir topun masanın herhangi bir noktasına düşme olasılığı eşittir. Bayes, bu düzlem üzerine ilk topun rastgele atıldığını, fakat topun nereye düştüğünü kendisinin bilmediğini varsayar. Ardından ikinci bir top atılır ve sadece bu topun ilk topun sağına mı yoksa soluna mı düştüğü bilgisi Bayes'e iletilir. Bu işlem defalarca tekrarlanır. Her yeni atışta, topun konumu yine sadece "ilk topun sağına mı soluna mı" düştüğü biçiminde raporlanır. Bayes, ilk topun (bilinmeyen nedenin) konumunu giderek daralan bir olasılık aralığına sıkıştırmaya başlar. Örneğin, eğer tüm toplar ilk topun sağına düşerse, Bayes bu ilk topun muhtemelen en sol uçta yer aldığını varsayar. Bu deneysel düşünce, Bayes'in şimdiki gözlemlerden geçmiş nedene ulaşma fikrini nasıl yapılandırıldığını gösterir. Bayes hiçbir zaman ilk topun tam yerini bilemez; ancak artan sayıda gözlemlerle, bu konum hakkında yüksek güvenle tahmin yapılabilir.

Bayesci yaklaşımın kavramsal yapısı oldukça basittir. Başlangıç inançları, güncel gözlem verileri ile birleştirilerek daha iyi bir inanca ulaşılır. Bu yapı şu şekilde formüle edilir:

$$Yeni\ İnanç\ (Posterior) = Önceki\ İnanç\ (Prior) \times Olabilirlik(Likelihood)$$

Sistemin her adımında bu güncelleme tekrarlanır; elde edilen her yeni sonsal dağılım (posterior), bir sonraki güncellemede önsel dağılım (prior) olarak kullanılır. Bu yönüyle Bayesçi yaklaşım, her yeni bilgiyle birlikte kendini iyileştiren ve doğruluğa giderek daha çok yaklaşan evrimsel bir çıkarım sistemidir (Gelman vd. 2013). Bu yaklaşım, bugün Bayes teoremi ile ifade edilen önsel bilginin (prior) yeni veri ile güncellenmesi (posterior) fikrinin sezgisel temelini oluşturur (McGrayne, 2011).

Bayes'in bu düşünce deneyini daha iyi kavrayabilmek için, bir örnek senaryo üzerinden hareket edilebilir. Masanın $[0, 1]$ aralığında doğrusal tek boyutlu bir düzlem olarak modellendiği varsayalım. Bu masaya ilk top rastgele atılsın ve konumu gözlemciden gizlensin. Örneğin, bu topun konumu gerçekte $\theta = 0.73$ noktasına denk gelmiş olsun. Ardından 10 farklı top konumları eşit olasılıkla $[0, 1]$ aralığına düşecek şekilde atılsın. Her atış sonrası gözlemciye yalnızca yeni atılan top ilk topun sağında mı yoksa solunda mı düştüğü bilgisi verilsin. Atışların sonucunda toplardan 7'si ilk topun soluna, 3'ü sağına düşmüş olsun.

Bu noktadan sonra gözlemcinin amacı, bu kısıtlı bilgidен yola çıkarak ilk topun konumu hakkında çıkarımda bulunmaktır. Başlangıçta, ilk topun masa üzerindeki konumu hakkında hiçbir bilgi olmadığı varsayılır; yani önsel dağılım (prior) en tarafsız ve bilgi içermeyen varsayım olarak $\theta \sim Uniform(0,1)$ kabul edilir. Ancak 10 gözlem sonucunda elde edilen “sol/sağ” verileri ile bu bilgi güncellenebilir.

Bayes Teoremi uygulanarak elde edilen sonsal dağılım (posterior), aşağıdaki şekilde modellenir:

$$p(\theta|veri) \propto \theta^7 \cdot (1 - \theta)^3 \quad (2.1)$$

Bu ifade, parametrik olarak $Beta(8,4)$ dağılımına karşılık gelir. Yani Bayes'in gözlemler sonrası inancı artık üniform değildir. θ değeri büyük olasılıkla sonsal dağılımın ($Beta(8,4)$) tepe noktası olan 0.7 civarındadır. Gözlem sayısı arttıkça dağılım daha

sivrileşir ve belirli bir noktada yoğunlaşır. Böylece ilk topun konumu hakkında daha güvenilir bir aralık elde edilir.

Bu örnek, Bayes'in neden-sonuç ilişkisinde tek tek değerleri bilmeye gerek duymadan, gözlemlerin yönü üzerinden olasılık güncellemesi yapabileceğini göstermektedir. Bu güncelleme, modern Bayes teoreminin temel mantığını sezgisel olarak temsil etmektedir.

Bayes'in bu düşünce biçimini Pierre-Simon Laplace matematiksel çerçeveye genelleştirmiştir (McGrayne, 2011; Gelman vd., 2013). Şu şekilde ifade edilir:

(Ω, U, P) bir olasılık uzayı ve $A_1, A_2, \dots, A_n \in U$ bu uzayda tanımlı herhangi bir parçalanma olsun. Yani, bu olaylar birbirinden bağımsız, aynı anda gerçekleşmesi mümkün olmayan ve $\bigcup_{i=1}^n A_i = \Omega$ olan birbirinden farklı ayrık olaylardan oluşsun. Bu durumda, bu olasılık uzayında $P(B) > 0$ koşulunu sağlayan bir B olayı gerçekleştiğinde herhangi bir A_i olayının olasılığı

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{\sum_{j=1}^n P(A_j)P(B|A_j)}, \quad (i = 1, 2, \dots, n) \quad (2.2)$$

ile hesaplanır. Bu formül, Bayes Teoreminin ayrık olaylar için genelleştirilmiş hali olarak verilir.

$y = y_1, y_2, \dots, y_n$ gözlemi olasılık (yoğunluk) fonksiyonu $o(y)f$ $p(y|\theta)$ olan dağılımdan bir örneklem olmak üzere, θ parametresi için $\Pi(\theta)$ önsel $o(y)f$ göz önüne alınsın. Bayes teoremi kullanılarak sonsal $o(y).f$,

$$p(\theta|y) = \frac{p(\theta, y)}{p(y)} = \frac{p(y|\theta)\Pi(\theta)}{p(y)} = \frac{p(y|\theta)\Pi(\theta)}{\int p(y|\theta)\Pi(\theta)d\theta} \quad (2.3)$$

biçiminde elde edilir. Burada, payda da yer alan $\int p(y|\theta)\Pi(\theta)d\theta$ ifadesi, sonsal dağılımın normalleştirme sabitidir. θ parametresinin olabilirlik fonksiyonu $p(y|\theta)$ $o(y)f$ nin bir orantılı fonksiyonu olmak üzere,

$$L(\theta) \propto p(y|\theta) \quad (2.4)$$

biçiminde orantısal olarak ifade edilebilir. Buna göre sonsal $o(y)f$,

$$p(\theta|y) = \frac{L(\theta)\Pi(\theta)}{\int L(\theta)\Pi(\theta)d\theta} \quad (2.5)$$

biçiminde yazılır. Burada, paydada yer alan $p(y)$ marjinal dağılımı bir integraldir (kesikli dağılımlar için toplam). Bu integralin değeri sonlu olduğu sürece sonsal dağılım ile ilgili herhangi bir ilave bilgi içermez. Bu sebeple, $p(\theta|y)$

$$p(\theta|y) \propto L(\theta)\Pi(\theta). \quad (2.6)$$

biçiminde orantısal olarak yazılabilir. Bayesci yaklaşım, aslında elde var olan bilginin yeni bilgiler geldikçe nasıl güncelleneceğini ifade eder. Bir parametre için önsel dağılım, veri analizine başlamadan önce parametre hakkında kesin olmayan bilgileri içeren olasılık dağılımıdır. Önsel $o(y)f$ ile olabilirlik fonksiyonunun çarpımı parametre için sonsal $o(y)f$ verir. Parametre için sonsal dağılım belirlendikten sonra tüm çıkarımlar bu dağılım kullanılarak yapılabilir (Cengiz vd., 2012) .

Bayesci yaklaşıma aşağıdaki şekilde bir örnek verilebilir;

Bir bilim insanı, belirli bir yeni tedavinin hastalar üzerinde etkili olup olmadığını test etmek istemektedir. Klasik frekansçı istatistikte, hipotez testi sonucu tedavinin etkili olup olmadığına karar vermek için p -değerine bakılır. Ancak Bayesci yaklaşımda, tedavi etkisinin bir önsel olasılık dağılımı belirlenir ve yeni veriler elde edildikçe bu olasılık güncellenir.

Adım 1: Ön Bilgi (Prior) Bilim insanı tedavinin etkili olma ihtimalini %30 olarak değerlendirmektedir:

$$P(Tedavi Etkili) = 0.30, P(Tedavi Etkisiz) = 0.70 \quad (2.7)$$

Adım 2: Veri Toplama ve Olabilirlik Hesaplama Yapılan klinik denede 100 hastanın 60'ında iyileşme olduğu görülmüştür.

Eğer tedavi etkiliyse, hastaların iyileşme olasılığının %80 olduğu, eğer tedavi etkisizse, iyileşme oranının %50 olduğu bilinmektedir. Buna göre,

$$P(\text{İyileşme}|\text{Tedavi Etkili}) = 0.80, P(\text{İyileşme}|\text{Tedavi Etkisiz}) = 0.50$$

koşullu olasılıkları yazılabilir.

Adım 3: Bayes Teoremi ile Güncelleme Yeni verileri kullanarak, tedavinin gerçekten etkili olma olasılığı güncellenir:

$$P(\text{Tedavi Etkili}|\text{İyileşme}) = \frac{P(\text{İyileşme}|\text{Tedavi Etkili})P(\text{Tedavi Etkili})}{P(\text{İyileşme})} \quad (2.8)$$

Burada normalleştirme sabiti $P(\text{İyileşme})$ toplam olasılık kuralına göre,

$$\begin{aligned} P(\text{İyileşme}) &= P(\text{İyileşme}|\text{Tedavi Etkili})P(\text{Tedavi Etkili}) \\ &\quad + P(\text{İyileşme}|\text{Tedavi Etkisiz})P(\text{Tedavi Etkisiz}) \\ &= (0.80 \times 0.30) + (0.50 \times 0.70) = 0.59 \end{aligned} \quad (2.9)$$

olarak bulunur. Formülde yerine konulduğunda

$$P(\text{Etkili}|\text{İyileşme}) = \frac{(0.80 \times 0.30)}{(0.80 \times 0.30) + (0.50 \times 0.70)} = \frac{0.24}{0.59} \approx 0.41$$

elde edilir. Yeni veriler ışığında, tedavinin etkili olma olasılığı %30'dan %41'e yükselmiştir. Yani, Bayes güncellemesi ile bilim insanı, tedaviye olan güvenini artırmıştır.

Ancak Bayesci istatistikte, bazen bu normalleştirme katsayısını hesaplamak zor veya imkansız olabilir. Çoğunlukla, yüksek boyutlu ve karmaşık modellerde integral hesaplamalarında bu problem ortaya çıkar. Bu durumda MCMC gibi yöntemler, doğrudan sonsal dağılımdan örnekleme yaparak bir çözüm önerir. (Ekinci, 2019)



3. DAĞILIMLARDAN RASGELE SAYI ÜRETME

MCMC yöntemlerinden bahsetmeden önce rastgele sayı üretmenin temel taşları olan Ters Dönüşüm Yöntemi ve Kabul-Red yöntemlerinin mantığını anlamak gerekir.

3.1 Ters Dönüşüm Yöntemi

X rasgele değişkenin dağılım fonksiyonu F olarak verilsin ve bu dağılımdan rasgele sayı üretilmek istensin. Bu durumda, en yaygın kullanılan yöntem, F fonksiyonunun genelleştirilmiş tersi olan

$$F^{-}: (0,1) \rightarrow R$$
$$u \rightarrow F^{-}(u) = \inf\{x: F(x) \geq u\} \quad (3.1)$$

fonksiyonu ile $X = F^{-}(U)$ dönüşümünü kullanmaktır. U ile $(0,1)$ aralığında Düzgün dağılıma sahip olan bir rasgele değişken ifade edilir. Bu dönüşüm ile Düzgün dağılımdan elde edilen U rasgele sayıları F dağılımına dönüştürülmüş olur.

Örnek olarak; olasılık yoğunluk fonksiyonu;

$$f(x) = \begin{cases} 2x & , \quad 0 \leq x \leq 1 \\ 0 & , \quad d.d. \end{cases} \quad (3.2)$$

olan F dağılım fonksiyonuna ters dönüşüm yöntemi uygulansın. İlk olarak dağılım fonksiyonu

$$F(x) = \begin{cases} 0, & x < 0 \\ x^2, & 0 \leq x \leq 1 \\ 1, & x > 1 \end{cases} \quad (3.3)$$

olmak üzere $X = F^{-1}(U) = U^{1/2} = \sqrt{U}$ dönüşümü ile düzgün dağılımdan üretilen rasgele sayıların kökü alınarak F dağılımdan sayı üretilmiş olur (Öztürk ve Özbek, 2015). Ancak bu yöntem çok boyutlu dağılımlara uygulanamaz.

3.2 Kabul-Red Yöntemi

Kabul Red yöntemi John von Neumann tarafında 1950'lerde, karmaşık dağılımlardan örnekleme elde etmek amacıyla geliştirilmiş, doğrudan örnekleme yapmanın zor olduğu durumlarda kullanılan bir yöntemdir. Çalışma mantığı basitçe şu şekildedir:

Önce kolay örnekleme yapılabilecek bir dağılım $G(x)$ ve $o.(y).f$ $g(x)$ seçilir. Sonra, bu dağılımın hedef dağılımı olan $f(x)$ ile benzer olup olmadıkları test edilir. Eğer $g(x)$ eğrisi, bir M çarpanıyla $f(x)$ 'in üzerinde olacak şekilde ölçeklendirilebilirse, her yeni örneklem için bir kabul testi yapılır.

Formülasyonu şu şekildedir:

X rasgele değişkeni f $o.(y).f$ na sahip ve değer kümesi D ile ifade edilsin. g $o.(y).f$ na sahip V rasgele değişkeninin değer kümesi de D olsun ve burada önemli nokta bu dağılımdan V sayılarını üretmek kolay olsun. Ayrıca, $\forall x \in D$ için

$$f(x) \leq a \cdot g(x) \quad (3.4)$$

olacak şekilde a pozitif sabit değeri olsun. Bu koşullar altında aşağıdaki algoritma ile $o.(y).f$ ile verilen dağılımdan X rasgele sayıları üretilebilir;

1.Adım

Birbirinden bağımsız olarak V rasgele değeri g fonksiyonundan ve U rasgele değeri $U(0,1)$ dağılımından üretilir.

2.Adım

Eğer,

$$U \leq \frac{f(V)}{a \cdot g(V)} \quad (3.5)$$

oluyorsa $X = V$ olarak alınır ve 1.Adıma geri dönülür.

Örnek olarak olasılık yoğunluk fonksiyonu;

$$f(x) = \begin{cases} \frac{2}{\pi} \sqrt{x(2-x)} & , \quad 0 < x < 2 \\ 0 & , \quad d.y. \end{cases} \quad (3.6)$$

olan fonksiyon göz önüne alınsın. Bu fonksiyona benzer uygun bir $g(x)$ fonksiyonu da;

$$g(x) = \begin{cases} \frac{1}{2} & , \quad 0 < x < 2 \\ 0 & , \quad d.y. \end{cases} \quad (3.7)$$

biçiminde seçilsin. Bu durumda, $a = 2$ için

$$f(x) \leq 2 \times g(x) \quad (3.8)$$

$x \in (0,2)$ 'dir. O halde;

$U_1, U \sim U(0,1)$ dağılımından üretilip $V = 2 \times U_1$ hesaplanır. Daha sonra yine $U_2, U(0,1)$ dağılımından üretilir.

1.Adım $U_2 \leq \frac{2}{\pi} \sqrt{V(2-V)}$ koşulu sağlanır ise $X = V$ alınır ve yeni sayı üretmek için 1. Adıma dönülür.

Kabul-Red yöntemi, rastgele seçilen her örneği bağımsız olarak değerlendirir ve kabul/red mekanizması ile çalıştırır. Ancak bu yöntem yüksek boyutlu sistemlerde çok fazla örneği reddettiği için verimsizdir. İşte bu noktada, Markov zinciri devreye girerek örnekleri birbirine bağlı hale getirir ve daha az örnek reddederek daha verimli bir örnekleme süreci oluşturur.

4. MARKOV ZİNCİRİ VE MONTE CARLO YÖNTEMLERİ

4.1 Markov Zinciri

Markov zinciri, sistemin zamanla alabileceği durumları belirli geçiş olasılıklarıyla modelleyen ayrık zamanlı stokastik bir süreçtir. Bu süreçte, sistemin bir sonraki durumda bulunma olasılığı yalnızca şu anki duruma bağlıdır; geçmiş durumların doğrudan etkisi yoktur. Durum uzayı genellikle sonlu ya da sayılabilir sonsuz bir kümeden oluşur ve sistemin geçişleri bir geçiş olasılıkları matrisi ile tanımlanır. Markov zincirinin temel yapı taşları; sistemin alabileceği tüm durumları içeren durum kümesi, bu durumlar arasında geçişleri rastlantısal olarak tetikleyen mekanizmalar ve geçiş olasılıklarını belirten karesel geçiş matrisidir (Çelik, 2013).

$\{X_t, t \in T\}$ biçiminde tanımlanan rassal değişkenlerin dizisi göz önüne alınsın. x_t, X_t 'nin $t \in T$ anın da aldığı değeri gösterebilir. Rassal değişkenler dizisi $\{X_t, t \in T\}$ eğer aşağıdaki koşulu sağlıyorsa Markov zinciri olarak adlandırılır:

$$P(X_t = x_t | X_{t-1} = x_{t-1}, \dots, X_0 = x_0) = P(X_t = x_t | X_{t-1} = x_{t-1}) \quad (4.1)$$

(Öztürk ve Özbek, 2004). Bu süreçte sistemin alabileceği durumlar bir durum kümesi E ile tanımlanır. Her bir $i \in E$ durumundan $j \in E$ durumuna bir adımda geçiş olasılığı,

$$p_{ij} = P(X_{t+1} = j | X_t = i) \quad (4.2)$$

koşullu olasılığı ile ifade edilir. i durumundan j durumuna n adımda geçiş olasılığı

$$p_{ij}^n = P(X_{t+n} = j | X_t = i) \quad (4.3)$$

ile ifade edilir. Geçiş matrisi, elemanları geçiş olasılıkları olan stokastik bir matristir (Ekici, 2019). Bu nedenle, matrisin her bir satır toplamı 1'e eşit olmalıdır. Araştırmacı, Markov zincirini belirlenen bir başlangıç noktası ile başlatır. Sürecin başladığı anda

durumların olasılıklarına ilk olasılıklar denir. İlk olasılıkların oluşturduğu vektör π_0 ile gösterilir ve zincir, bir π_0 başlangıç vektörünün atanması ile başlar. Başlangıç vektörünün bir elemanı dışında diğer tüm elemanları 0 değerini alır. Örneklem uzayı $S = \{0,1,2, \dots\}$ biçiminde olan bir Markov zinciri için bir adım geçiş olasılıklarının oluşturduğu matris “geçiş matrisi” adını alır ve bu geçiş matrisi $P = [p_{ij}]$ ile gösterilir. Elemanları, n adım geçiş olasılıklarından oluşan matris ise $P^n = [p_{ij}^n]$ ile tanımlanır. Amaç, geçiş matrisinin durağan duruma ya da kararlı bir orana gitmesini sağlamaktır. Burada bahsedilen durağan durum ya da zincirin durağan dağılımı örneklemede kullanılacak önemli bir özelliktir.

4.2 Monte Carlo İntegrasyonu

Monte Carlo yöntemleri, karmaşık matematiksel işlemler için rastgele örnekleme yoluyla yaklaşık çözümler sunan sayısal tekniklerdir. Bu yöntemlerin özü, çözülmek istenen matematiksel problemin bir beklenen değer olarak ifade edilmesi ve bu beklentinin rastgele örnekleme yoluyla hesaplanmasıdır. Özellikle analitik olarak hesaplanması güç veya yüksek boyutlu integraller için kullanımı son derece etkilidir. Burada amaç;

$$\int_a^b h(x)dx \quad (4.4)$$

biçiminde verilen karmaşık yapıdaki integralin hesaplanmasıdır. Sınırları, a ve b olarak verilmiş bu aralıkta tanımlı olan bir $h(x)$ fonksiyonu, aşağıda verildiği gibi parçalanabilirse, Monte Carlo yaklaşımı ile bu integral hesaplanabilir,

$$\int_a^b h(x)dx = \int_a^b g(x)f(x)dx. \quad (4.5)$$

Eşitlikte verilen $g(x)$ fonksiyonu ile X rasgele değişkeninin bir fonksiyonu ve $f(x)$ ile ise rasgele değişkenin olasılık yoğunluk fonksiyonu ifade edilsin. Bu durumda, eşitliğin sağ tarafında yer alan integral bir beklenen değer olan $E_{f(x)}(g(x))$ ile ifade edilebilir. Burada asıl sorun, verilen integralin hesaplanmasının zor olmasıdır. Buna göre, eğer $f(x)$ olasılık

yoğunluk fonksiyonu kullanılarak rasgele değişkenin dağılımından örnekler çekilebilirse, integral aşağıdaki yaklaşık değere eşit olur,

$$\int_a^b h(x)dx \cong E_{f(x)}(g(x)) \quad (4.6)$$

Elde edilen bu eşitlik, Monte Carlo İntegrasyonu olarak adlandırılır (Demirci, 2016).

Tezin başından beri bahsedildiği gibi, karmaşık olasılık dağılımlarını elde etmek çoğu zaman analitik yöntemlerle mümkün değildir. Bu gibi durumlarda Monte Carlo yöntemleri, rastgele örnekleme gücünden yararlanarak dağılımı yaklaşık olarak incelemeyi sağlar. Ancak, hedef dağılımdan her seferinde bağımsız örnekler çekmek hem hesaplama açısından verimsizdir hem de dağılımın yapısal özelliklerini yeterince yansıtamayabilir. Bu sorunu aşmak için geliştirilen MCMC yöntemleri, örnekleme sürecini birbirine bağlı adımlardan oluşan bir Markov zinciri üzerinden yürütür. Her yeni örnek, bir önceki örneğe bağlı olarak üretilir. Bu şekilde kurulan örnekleme zinciri, zamanla hedef dağılıma yakınsayan bir yapı sergiler ve dağılımın karakteristik özelliklerini temsil eden örnekler üretir (Betancourt, 2017).

4.3 MCMC Yöntemlerinin Genel İşleyişi

MCMC yöntemlerinde temel amaç, belirli bir hedef dağılıma ($\pi(\theta)$) uygun örnekler üretebilecek bir Markov zinciri oluşturarak bu dağılım üzerinden istatistiksel çıkarımlar yapaktır (Robert ve Casella, 2004).

Öncelikle modelin yapısına bağlı olarak hedef dağılımın tanımlanması gerekir. Bayesci yaklaşımla çalışıldığında hedef dağılım genellikle sonsal dağılımlardır. Ardından zincirin başlatılabilmesi için parametreler için uygun başlangıç noktalarının belirlenmesi gerekir. Zincirin başlangıç değeri, modelin önsel bilgilerine dayanarak, örneğin önceki araştırmalarda kullanılan başlangıç değerleri olarak veya rastgele belirlenebilir. Ancak başlangıç noktasının hedef dağılımın karakteristik bölgelerine (tipik kümeye) yeterince yakın olması, yakınsama sürecinin hızlanması açısından önemlidir.

Uygun MCMC algoritmasının seçilmesi, örnekleme sürecinin verimliliği açısından önemlidir. Basit yapılarda Metropolis-Hastings ya da Gibbs örnekleyici tercih edilirken, yüksek boyutlu ve daha karmaşık modellerde Hamiltonian Monte Carlo (HMC) gibi daha etkin sonuçlar veren yöntemlere başvurulabilir.

Seçilen algoritma doğrultusunda bir geçiş mekanizması tanımlanır. Bu mekanizma, zincirin her adımında bir sonraki örneğin nasıl üretileceğini belirler. Üretilen yeni örneklerin hedef dağılım altında kabul edilip edilmeyeceği, algoritmaların kabul kriterlerine göre yapılır.

Zincir, başlangıç döneminde hedef dağılıma henüz ulaşılmamış olabilir. Bu nedenle ilk birkaç yüz ya da bin iterasyon genellikle “ısınma” (burn-in) dönemi olarak kabul edilir ve analizlerde dikkate alınmaz. Isınma döneminden sonra elde edilen örnekler, hedef dağılımın yapısını yansıttığı varsayımıyla istatistiksel hesaplamalarda kullanılır (Betancourt, 2017).

Örnekleme sürecinin başarısı; zincirin karışım kalitesi, otokorelasyon düzeyi, dağılımın tüm bölgelerine erişebilme kapasitesi ve özellikle tipik küme olarak adlandırılan yoğunluk ve hacmin dengelendiği bölgeyi ne derece temsil edebildiğiyle doğrudan ilişkilidir. Bu nedenle MCMC uygulamalarında zincirin yakınsayıp yakınsamadığı, örneklerin bağımsızlığı ve örnekleme verimliliği gibi faktörler çeşitli tanılayıcı grafik ve metriklerle (örneğin iz grafiği (trace plot), otokorelasyon fonksiyonu, Gelman-Rubin $\hat{R}$ istatistiği) dikkatle değerlendirilmelidir (Betancourt, 2017).

Literatürde yer alan MCMC yöntemleri devam eden alt bölümlerde verilmiştir.

4.3.1 Metropolis ve Metropolis-Hastings algoritması yöntemi

MCMC yöntemlerinden en bilinenleri Metropolis ve onun genelleştirilmiş hâli olan Metropolis-Hastings algoritmalarıdır. Özellikle çok boyutlu, normalize edilmesi zor, çok modlu veya yüksek korelasyonlu veri yapılarında sıklıkla tercih edilirler. Basit yapılarda

ve öneri dağılımının simetrik seçilebildiği durumlarda Metropolis algoritması yeterlidir. Öneri dağılımı asimetrikse veya veri yapısına özel geçiş stratejileri belirlenmişse, Metropolis-Hastings algoritması tercih edilir. Bayesci yaklaşımda, sonsal dağılımlardan örneklem çekmek için bu algoritmalar kullanılır. Bu bölümde öncelikle Metropolis sonrasında bu algoritmanın genelleştirilmiş şekli olan Metropolis-Hastings algoritmalarından bahsedilecektir.

Hedef dağılımı olarak $\pi(\theta)$ tanımlanmış olsun. Bu dağılım genellikle normalize edilmemiş bir sonsal dağılımdır. Doğrudan $\pi(\theta)$ 'dan örnekleme yapmak zordur. Bu nedenle θ 'nın bağlı olduğu bir olasılık yoğunluk fonksiyonu olan $f(\theta)$ üzerinden örnekleme yapılır. $f(\theta)$ 'nın beklenen değeri gibi özellikleri hesaplanmak isteniyorsa bu fonksiyon şu şekilde olmalıdır:

$$f(\theta) = \frac{\pi(\theta)}{c} \quad (4.7)$$

burada c , normalleştirme katsayısı olarak bilinmektedir. Algoritmaya, rastgele ya da sezgisel olarak $\pi(\theta_0) > 0$ koşulunu sağlayan bir θ_0 başlangıç değeri ile başlanır. i . iterasyonda, bir önceki iterasyon için geçerli θ_{i-1} değeri göz önüne alınarak $q(\theta^*|\theta_{i-1})$ öneri dağılımından θ^* aday noktası çekilir. Öneri dağılımı Metropolis Algoritması için $N(\theta_{i-1}, \sigma^2)$ gibi simetrik olmalıdır. Yani $[q(\theta_i)|\theta_{i-1}] = q(\theta_{i-1}|\theta_i)$ olmalıdır. Aday noktanın kabul olasılığı şu şekilde hesaplanır:

$$\alpha = \min\left(\frac{\pi(\theta^*)}{\pi(\theta_{i-1})}, 1\right) \quad (4.8)$$

Bu noktada, $U(0,1)$ dağılımından $U \sim U(0,1)$ rasgele bir sayı çekilir. Eğer $u < \alpha$ ise aday nokta θ^* bir sonraki iterasyonun başlangıç değeri olur. Eğer $u > \alpha$ ise θ^* reddedilir ve yeni bir aday nokta belirlenir.

Bu algoritma ile θ_i 'den θ_{i+1} 'e geçiş olasılıkları sadece θ 'ya bağlı olan $(\theta_0, \theta_1, \dots, \theta_m, \dots)$ biçiminde bir Markov zinciri oluşturulur. Başlangıç değerlerinin etkisini kaybetmesi için yeterli iterasyon sayısına ulaşıldıktan sonra (m ile yeterli

iterasyon sayısı gösterilir) zincir durağan dağılıma yaklaşır. Bu şekilde, $(\theta_0, \theta_1, \dots, \theta_m, \dots)$ ile elde edilen örneklemeler $\pi(\theta)$ dağılımından üretilen örneklemelere karşılık gelir.

Hastings, Metropolis algoritmasını keyfi bir geçiş olasılığı fonksiyonu kullanarak genelleştirmiştir. Hastings algoritmasında, noktanın kabul edilme olasılığı

$$\alpha = \min \left(\frac{q(\pi(\theta^*)f(\theta_{i-1}|\theta^*))}{\pi(\theta_{i-1})q(\theta^*|\theta_{i-1})}, 1 \right). \quad (4.9)$$

biçiminde tanımlamıştır. Bu kabul olasılığını ve geçiş dağılımını kullanarak oluşturulan algoritma Metropolis-Hastings Algoritması olarak adlandırılır. Metropolis algoritmasından farkı, öneri dağılımı üzerine herhangi bir koşul olmamasıdır, öneri dağılımının simetrik olması gerekli değildir.

MCMC yöntemleri uygulanırken, tek bir uzun zincirin oluşturulması ya da farklı başlangıç değerlerine sahip birden fazla zincirin paralel olarak çalıştırılması gibi farklı stratejiler benimsenebilir. Her ne kadar tek bir uzun zincir kullanmak teorik olarak hedef dağılıma yakınsamak için yeterli olsa da pratikte bu yaklaşım başlangıç değerlerinin etkisini ortadan kaldırmak için uzun bir geçiş (burn-in) süresi gerektirebilir. Ayrıca, özellikle yüksek otokorelasyonun gözlemlendiği durumlarda birden fazla bağımsız zincir çalıştırmak daha güvenilir sonuçlar üretebilir. Bu tür çoklu zincir stratejileri, modelin hedef dağılım üzerindeki keşif kapasitesini artırır ve hedef dağılıma yakınsamak için avantaj sağlayabilir (Brooks vd., 2011).

Çok modlu hedef dağılımlarda uygulanan Metropolis veya Metropolis Hastings Algoritmalarının oluşturduğu Markov zinciri bu modlardan birinde sıkışma eğiliminde olabilir. Geyer (1991) bu tür durumlarda, birden fazla zincirin eş zamanlı kullanıldığı gelişmiş bir algoritma olan Paralel Tempering yöntemini önermektedir. Bu yöntemde farklı sıcaklıklara (T) karşılık gelen farklı zincirler çalıştırılır. Burada sıcaklık, dağılımın keskinliğini kontrol eder; sıcaklık arttıkça dağılım düzleşir ve zincirin yüksek olasılıklı

bölgeler arasında daha kolay geçiş yapmasına olanak tanır (Earl ve Deem, 2005). Her bir zincir, hedef dağılımın $\pi_T(x)$ 'in şu şekilde ölçeklenmiş versiyonunu örnekler:

$$\pi_T(x) \propto \pi(x)^{1/T} \quad (4.10)$$

Geyer'in önerdiği bu yöntemde, farklı “sıcaklık” seviyelerine karşılık gelen birden fazla zincir paralel olarak çalıştırılır. Sıcaklık parametresi $T > 1$ olan dağılımlar daha düzleşmiş yani modlar arası geçişi kolaylaştıran yapıya sahiptir. Belirli sayıda iterasyon tamamlandıktan sonra, farklı sıcaklıklarda çalışan iki zincir rastgele seçilir. Bu zincirlerden biri sıcaklık T_i ile çalışan ve mevcut durumda x_i konumunda bulunan zincir; diğeri ise sıcaklık T_j ile çalışan ve mevcut durumda x_j konumunda bulunan zincir olsun. Bu iki zincirin durumlarının karşılıklı olarak değiştirilmesi (swap) önerilir. Bu değişimler Metropolis-Hastings algoritmasında olduğu gibi bir kabul-red mekanizması kullanılarak uygulanır.

$$\alpha = \min\left(\frac{\pi(x_j)^{1/T_i} \pi(x_i)^{1/T_j}}{\pi(x_i)^{1/T_i} \pi(x_j)^{1/T_j}}, 1\right) \quad (4.11)$$

Burada α değeri rassal olarak elde edilen bir u ($U \sim Uniform(0,1)$) değeri ile karşılaştırılır. Eğer $u < \alpha$ ise zincirler arası değişim (swap) kabul edilir. Değişim kabul edildiğinde farklı sıcaklıklarda çalışan iki zincirin yalnızca mevcut durumları (x_i ve x_j) yer değiştirir. Böylece, orijinal hedef dağılımı örnekleyen zincirin keşif kapasitesi artırılır (Geyer, 1991; Wilkinson, 2013).

4.3.2 Gibbs örnekleme yöntemi

Metropolis-Hastings Algoritmasının özel bir durumu olan Gibbs örnekleme, sonsal dağılımın tam koşullu dağılım serisinin oluşturulabildiği durumlarda kullanılır (Gamerman ve Lopes, 2006). Gibbs örneklemeinde temel amaç, koşullu ilişkileri kullanarak tahmin sürecini ardışık olarak sürdürmek ve parametrelerin bileşik dağılımlarına ulaşabilmektir. Algoritma şu şekildedir:

$(\theta = \theta_1, \theta_2, \dots, \theta_k, \dots)$ olmak üzere ilgilenilen parametreye ait dağılımın $\pi(\theta)$ olsun. θ vektörüne ait tüm tam koşullu dağılımların kümesi ise Θ ile gösterilsin. Burada θ 'nın k tane bileşeninin her biri sabit vektör ya da matris biçiminde çok boyutlu olabilir. Bu durumda θ için tam koşullu dağılımlar kümesi

$$\pi(\Theta) = \pi(\theta_i | \theta_{-i}) \quad (4.12)$$

biçiminde ifade edilmektedir. Burada, θ_{-i} tam koşullu dağılımlar kümesinin θ_i 'yi içermediği halidir. Gibbs örneklemesinde tanımlanan dağılımlardan örneklem çekilebilmek için, ilgilenilen parametrelere ilişkin tüm tam koşullu dağılımların analitik olarak tanımlanabilir biçimde ve tamamen birbirlerine bağımlı olması gerekir. Bu şekilde tam koşullu dağılımlardan birbirini izleyen alternatif bir veri türetme planı uygulanarak tahmin işlemine geçilebilir. İterasyon sayacı j ile ifade edilmek üzere, Gibbs örnekleme adımları aşağıdaki gibidir:

1.Adım Parametrelerin başlangıç değerleri belirlenir.

2.Adım

$$\theta^0 = (\theta_1^0, \theta_2^0, \dots, \theta_k^0) \quad (4.13)$$

3.Adım j . iterasyonda k adet parametreye ilişkin tam koşullu dağılımlar kümesi oluşturulur ve bu dağılımlar kullanılarak birbirini izleyen değerler çekilip j . döngü (zincir) tamamlanır. k tane parametreye ilişkin tam koşullu dağılımlar kümesi

$$\begin{aligned} \theta_1^j &\sim \pi(\theta_1 | \theta_2^{j-1}, \theta_3^{j-1}, \dots, \theta_k^{j-1}) \\ \theta_2^j &\sim \pi(\theta_2 | \theta_1^j, \theta_3^{j-1}, \dots, \theta_k^{j-1}) \\ \theta_3^j &\sim \pi(\theta_3 | \theta_1^j, \theta_2^j, \theta_4^{j-1}, \dots, \theta_k^{j-1}) \\ &\vdots \\ &\vdots \\ &\vdots \end{aligned} \quad (4.14)$$

$$\theta_j^j \sim \pi(\theta_j | \theta_1^j, \theta_2^j, \dots, \theta_{j-1}^j, \theta_{j+1}^{j-1}, \dots, \theta_k^{j-1})$$

·
·
·

$$\theta_k^j \sim \pi(\theta_k / \theta_1^j, \theta_2^j, \dots, \theta_{k-1}^j)$$

biçimindedir.

4.Adım Yakınsama sağlanana kadar her iterasyon için j , 1 artırılarak devam edilir ve tekrar 2. adıma dönülerek döngü (zincir) oluşturulmaya devam edilir.

İterasyonun 2. Adımında dikkat edilirse, tam koşullu dağılımlar kümesi birbirini izleyen değerlerin üretimi ile elde edilir. Kümede yer alan ilk bileşen ilk satırda elde edilir, daha sonra, yeni değeri bir sonraki adımda yer alan ikinci bileşenin elde edilmesinde kullanılır. Burada, parametreler arasındaki sıralama önemli değildir fakat diğer örneklemeler arasında en güncel olanının tercih edilmesi gerekir. Gibbs örnekleme için yazılan algoritma yeterince uzun çalıştırıldığında, var olan tüm döngüler koşullu dağılımları kullanarak, θ 'nın bütün bileşenleri için birer örneklem türetebilir. Fakat, bu örneklemenin çalışabilmesi için θ parametrelerinin tüm koşullu dağılımları kolayca elde edilebilir olmalıdır. Eğer elde edilemiyorsa, parametreler için Metropolis-Hasting algoritması kullanılarak güncelleştirme yapılır (Demirci, 2016).

Bayesci yaklaşım, özellikle sınıflandırma problemlerinde belirsizlik altında karar verme süreçlerinde etkin biçimde kullanılmaktadır. Bu kapsamda, Resnik ve Hardisty (2012) tarafından sunulan “Gibbs Sampling for the Uninitiated” başlıklı çalışmada, Gibbs örnekleme yöntemi kullanılarak bir e-postanın spam olup olmadığına ilişkin tahminler Bayesci çerçevede gerçekleştirilmiştir. Bu kısımda, söz konusu çalışmadan esinlenerek, sınıflandırma problemlerinde Bayesci modelleme ve MCMC yaklaşımının nasıl birlikte kullanılabileceği sadeleştirilmiş bir örnekle verilmiştir.

Problem Tanımı:

Bazı kelimeler spam e-postalarda daha sık görülür. Amaç, her bir e-postanın spam olup olmadığını, içerdiği kelimelere göre tahmin etmektir. Bu doğrultuda model aşağıdaki biçimde kurulur:

- $z_i \in \{0,1\}$: i . e-postanın sınıfı; spam ($z_i = 1$) veya değil ($z_i = 0$)
- $\theta = P(z = 1)$: herhangi bir e-postanın spam olma olasılığı
- $x_i = (x_{i1}, x_{i2}, \dots, x_{ij})$: i . e-postadaki kelimelerin var-yok durumu (ikili-binary)
- $\phi_{jk} = P(x_j = 1 | z = k)$: j ci kelimenin sınıf k iken görülme olasılığı

Önsel Dağılımlar:

Modeldeki belirsizlikleri ifade etmek için şu önsel dağılımlar kullanılır:

$$\theta \sim \text{Beta}(\alpha, \beta)$$

$$\phi_{jk} \sim \text{Beta}(a, b) \quad (\text{tüm kelimeler } (j) \text{ ve sınıflar } (k = 0,1) \text{ için}) \quad (4.15)$$

Bu yapı, parametrelerin değerleri hakkında gözlem öncesi inançları tanımlar. Burada kullanılan $\text{Beta}(\alpha, \beta)$ önsel dağılımı, parametrelerin olasılık değerleri hakkındaki başlangıçtaki inançları temsil eder. Beta dağılımının parametreleri olan α ve β , sırasıyla geçmişte gözlenmiş veya varsayılmış “başarı” ve “başarısızlık” sayılarının bir fazlasını temsil eder. Örneğin, θ için $\text{Beta}(1,1)$ seçimi, herhangi bir ön bilgiye sahip üniform bir önsel dağılımdır.

4.3.2.1 Sonsal dağılım ve Gibbs örnekleme

Veriler gözlemlendiğinde, eldeki e-posta örneklerinin hangi sınıfa ait olduğu (spam ya da ham) doğrudan bilinemez. Bu nedenle, Bayesci yaklaşımla her e-postanın spam olma olasılığı tahmin edilir ve buna göre sınıf etiketi örneklenir. Gibbs örnekleme, her parametre için koşullu olasılıklardan ardışık örnekleme yaparak bu süreci iteratif şekilde yürütür. Adımlar şu şekildedir:

Sınıf etiketleri (z_i) için örnekleme:

Her e-posta için, spam olma olasılığı şu şekilde hesaplanır:

$$P(z_i = 1|x_i, \phi, \theta) \propto \theta \cdot \prod_{j=1}^J \phi_{j1}^{x_{ij}} (1 - \phi_{j1})^{1-x_{ij}} \quad (4.16)$$

Bu olasılık, bir Bernoulli dağılımı üzerinden örneklenerek z_i değeri atanır (yani spam mi, değil mi kararı verilir).

θ parametresi için örnekleme:

Tüm e-postalar içinde kaç tanesinin spam olarak işaretlendiğini (n_1) sayarak ($n_1 = \sum z_i$), şu güncellenmiş beta dağılımından örnekleme yapılır:

$$P(\theta|z) \sim \text{Beta}(\alpha + n_1, \beta + N - n_1) \quad (4.17)$$

ϕ_{jk} parametreleri için örnekleme:

Her bir kelime (j) ve sınıf ($k \in \{0,1\}$) için, o kelimenin ilgili sınıfta geçme olasılığı şu beta dağılımından örneklenir:

$$\phi_{jk} \sim \text{Beta}(a + \sum_{i:z_i=k} x_{ij}, b + \sum_{i:z_i=k} (1 - x_{ij})) \quad (4.18)$$

Bu üç adım ardışık olarak birçok kez tekrarlandığında, parametrelerin ve gizli sınıf etiketlerinin sonsal dağılımlarından örnekler elde edilir. Böylece hem sınıflandırma problemi (spam tahmini) çözülmüş olur hem de parametre tahminleri belirsizlikleriyle birlikte yapılmış olur.

Bu örnek, özellikle sınıflandırma problemlerinde Bayesçi yaklaşımın sezgisel gücünü ve Gibbs örneklemesinin basit ama etkili yapısını göstermektedir. Modelin esnekliği, önsel dağılımlar yoluyla bilgi dahil edilmesini ve sonsal tahminlerin dağılımsal özellikleriyle değerlendirilmesini sağlar.

4.3.3 Hamiltonian Monte Carlo yöntemi

Klasik MCMC yöntemleri yüksek boyutlu ya da güçlü korelasyon içeren parametre uzaylarında yavaş yakınsama, yüksek otokorelasyon ve zayıf keşif gibi sorunlarla karşılaşabilmektedir. Her adımda rastgele küçük bir zıplama yapılır, ama bu zıplamalar çok yavaş dağılımı keşfeder. Hamiltonian Monte Carlo (HMC); Hamiltonian mekaniğinden ilham alarak örnekleme sürecini bir parçacığın potansiyel ve kinetik enerjiden oluşan toplam enerjisini (Hamiltonian) koruyarak hareket etmesi olarak yorumlar.

Hamilton Mekaniği Newton'un hareket yasasına enerji temelli bir yaklaşımdır. Bu yaklaşımda q konumu, p momentumu temsil eder, sistemin toplam:

$$H(q, p) = U(q) + K(p) \quad (4.19)$$

Burada $U(q)$ potansiyel enerji, $K(p)$ kinetik enerjidir ($K(p) = \frac{1}{2}p^T M^{-1}p$). Bu H fonksiyonuna Hamilton fonksiyonu denir. Fizikte bu enerjiyi kullanarak bir parçacığın nasıl hareket ettiği hesaplanır. Hamilton'un mekaniği bize şu denklemi verir:

$$\frac{dq}{dt} = \frac{\partial H}{\partial p}, \quad \frac{dp}{dt} = -\frac{\partial H}{\partial q} \quad (4.20)$$

Bu denklemden; konum, momentuma göre nasıl değişir ve momentum, konuma göre nasıl değişir bilgileri alınır. Yani parçacığın zaman içindeki rotası bulunur. Enerji korunur q konumu küçük ama tutarlı adımlarla değişir, yön değişimleri ani değildir ve geri zıplamaz. Mesela bir top dağın yamacına konulsun, topun hareketi sürekli ve eğimin yönüne göre devam eder. Geriye dönüş olmaz.

HMC yöntemi bu mekaniği kullanarak örnekleme zincirini model üzerinde uzak noktalara da götürür ama yine de hedef dağılımda yüksek olasılıklı bölgede kalır, böylece hedef dağılımı iyi temsil eden verimli bir zincir üretir.

q konumu örneklenmek istenen parametre vektörü θ olsun ve eklenen rastgele bir momentum vektörü p de hızı (momentumu) temsil etsin. Hedef dağılım $\pi(\theta)$ 'nın negatif logaritması potansiyel enerji olarak tanımlanır:

$$U(\theta) = -\log \pi(\theta) \quad (4.21)$$

Momentum değişkeni p genellikle çok değişkenli bir Normal dağılımdan ($p \sim N(0, M)$) çekilir ve kinetik enerji şu şekilde tanımlanır

$$K(p) = \frac{1}{2} p^T M^{-1} p. \quad (4.22)$$

Toplam enerji (Hamiltonian) fonksiyonu şu şekildedir

$$H(\theta, p) = U(\theta) + K(p). \quad (4.23)$$

Örnekleme süreci, Hamiltonian mekaniğinde tanımlanan aşağıdaki diferansiyel denklemlerin sayısal çözümüne dayalıdır

$$\frac{d\theta}{dt} = \frac{\partial H}{\partial p}, \quad \frac{dp}{dt} = -\frac{\partial H}{\partial \theta}. \quad (4.24)$$

Bu denklemler, enerji korunumunu sağlayacak şekilde bilgisayarda yaklaşık çözüm üretmek için Leapfrog adı verilen özel bir hesaplama yöntemi ile çözülür. Belirli bir süre (adım sayısı ve adım büyüklüğü belirlenerek) sistem simüle edilir ve yeni bir öneri noktası (θ^*, p^*) elde edilir. Bu öneri noktası Metropolis benzeri bir kabul-red mekanizması ile kabul edilir veya reddedilir:

$$\alpha = \min(1, \exp(H(\theta, p) - H(\theta^*, p^*))) \quad (4.25)$$

Bu yapı sayesinde, HMC örnekleri hedef dağılımın yüksek yoğunluklu bölgelerinde daha geniş adımlarla, otokorelasyonu düşük bir şekilde üretebilir. Özellikle yüksek boyutlu ve

karmaşık yapıllı modellerde, HMC algoritmasının yakınsama hızı ve örnekleme kalitesi klasik algoritmalara kıyasla oldukça yüksektir (Betancourt, 2017).

Adımları aşağıdaki gibidir:

1.Adım Parametrelerin başlangıç değerleri belirlenmelidir

$$\theta^0 = (\theta_1^0, \theta_2^0, \dots, \theta_k^0) \quad (4.26)$$

2.Adım Yardımcı momentum p vektörü oluşturulur

3.Adım

$$p^0 = (p_1^0, p_2^0, \dots, p_k^0), p^0 \sim N(0, I) \quad (4.27)$$

(burada I birim kovaryans matrisidir)

4.Adım Hamilton dinamiğine bağlı yörünge takibi $H(\theta, p) = U(\theta) + K(p)$ fonksionuyla L adım boyunca (θ, p) çifti aşağıdaki gibi güncellenir: (önce p yarım adım, θ tam adım, sonra p tekrar yarım adım güncellenir-leapfrog'un yarım-adım tekniği)

$$\begin{aligned} p_{t+\frac{1}{2}} &= p_t - \frac{\epsilon}{2} \nabla U(\theta_t) \\ p_t &= p_t + \epsilon M^{-1} \theta_{t+\frac{1}{2}} \\ p_{t+1} &= p_{t+\frac{1}{2}} - \frac{\epsilon}{2} \nabla U(\theta_{t+1}) \end{aligned} \quad (4.28)$$

Bu adımlar bizim belirlediğimiz toplam L kez tekrarlanarak (θ^*, p^*) çifti elde edilir. (burada ϵ : adım boyu sabiti, $\nabla U(\theta_t)$: potansiyel enerjinin gradyanı)

5.Adım Yeni önerilen θ^* değeri Hamilton fonksiyonu üzerinden Metropolis-Hastings kabul kriterine göre değerlendirilir.

$$\alpha = \min(1, \exp(H(\theta, p) - H(\theta^*, p^*))) \quad (4.29)$$

Belirlenen bu kabul olasılığına göre θ^* değeri kabul edilir veya mevcut değer korunur.

Momentum vektörü p geçici olarak kullanıldığından her iterasyon sonunda silinir ve bir sonraki iterasyonda yeni rastgele momentum vektörü örneklenir.

Yakınsama sağlanana kadar her iterasyonda yukarıdaki adımlar tekrarlanarak yeni örnekler elde edilir.

3. adımda θ parametresi Leapfrog adımlarıyla yeni bir θ^* değerine doğru ilerlemektedir. Burada seçilen L iterasyon sayısı eğer küçükse (yani az adım atılırsa) θ^* noktası başlangıç noktası θ 'ya çok yakın olur. Zincir küçük adımlarla ilerler. Bu da zincirin yüksek otokorelasyonlu olmasına neden olur. Eğer L iterasyon sayısı büyükse θ^* noktası başlangıç noktası θ 'dan çok daha uzak bir noktada olur. Zincir daha geniş bir bölgeyi keşfeder ancak bu durum simülasyon hızını düşürür. Bu sorunu çözmek amacıyla geliştirilen No-U-Turn Sampler (NUTS) algoritması, Hamilton dinamikleri altında oluşturulan örnekleme yolu boyunca hareketi takip eder. Eğer bu yol, geri dönme eğilimi gösterirse (yani yeni noktalar, başlangıç noktasına yönelirse), algoritma örnekleme sürecini otomatik olarak durdurur. Böylece örnekleme verimli bir şekilde gerçekleştirilir, gereksiz hesap yükü önlenmiş olur.

NUTS algoritmasına göre; yukarıda 3. Adımda bahsedilen Leapfrog yöntemiyle (θ, p) çiftinden (θ^*, p^*) çiftinin elde edilmesinde iterasyon $(p^* - p)^T(q^* - q) < 0$ durdurulur. Bu adım içerisinde θ^* için hesaplanan değerlerden biri rastgele atanır. Böylece L iterasyon sayısı yeterince uzun tutulmuş olur (Hoffman ve Gelman, 2014).

5. MCMC YÖNTEMİ İLE DAĞILIMLARDAN SAYI ÜRETME UYGULAMALARI

Tezin bu bölümünde bir önceki bölümde anlatılan MCMC yöntemleri kullanılarak bilinen dağılımlardan sayı üretme ile ilgili çeşitli örnekler yapılmıştır.

MCMC yöntemleri uygulanırken, iki özelliğe dikkat edilmelidir.

1) Yakınsama: $\theta^{(t)}$ nin dağılımı ne kadar çabuk $P(\theta|x)$ e yaklaşır.

MCMC zincirinin hedef dağılıma gerçekten ulaşp ulaşmadığı kontrol edilmelidir. Bunun için, trace plot (iz grafiği) çizdirilebilir. Yakınsamanın gerçekleşmesi için iz grafiğinin düz ve istikrarlı görünmesi gerekir, zamanla sistemli bir artma ya da azalma gözlenmemelidir.

2) Etkinlik: $\theta^{(t)}$ kullanılarak $P(\theta|x)$ için ne kadar iyi tahmin yapılır.

MCMC analizlerinde etkinlik, elde edilen örneklerin istatistiksel olarak ne kadar bilgi taşıdığı ile ilgilidir. Özellikle örnekler yüksek otokorelasyonlu olduklarında, örnek sayısı çok olsa bile “az bilgi” içerebilir. Bu yüzden MCMC zincirlerinde etkili örnek sayısı önemli bir konudur.

5.1 Cauchy Dağılımından Sayı Üretmek İçin Metropolis Algoritması

Standart Cauchy(0,1) dağılımının olasılık yoğunluk fonksiyonu

$$f(\theta) = \frac{1}{\pi(1+\theta^2)}, -\infty < \theta < \infty \quad (5.1)$$

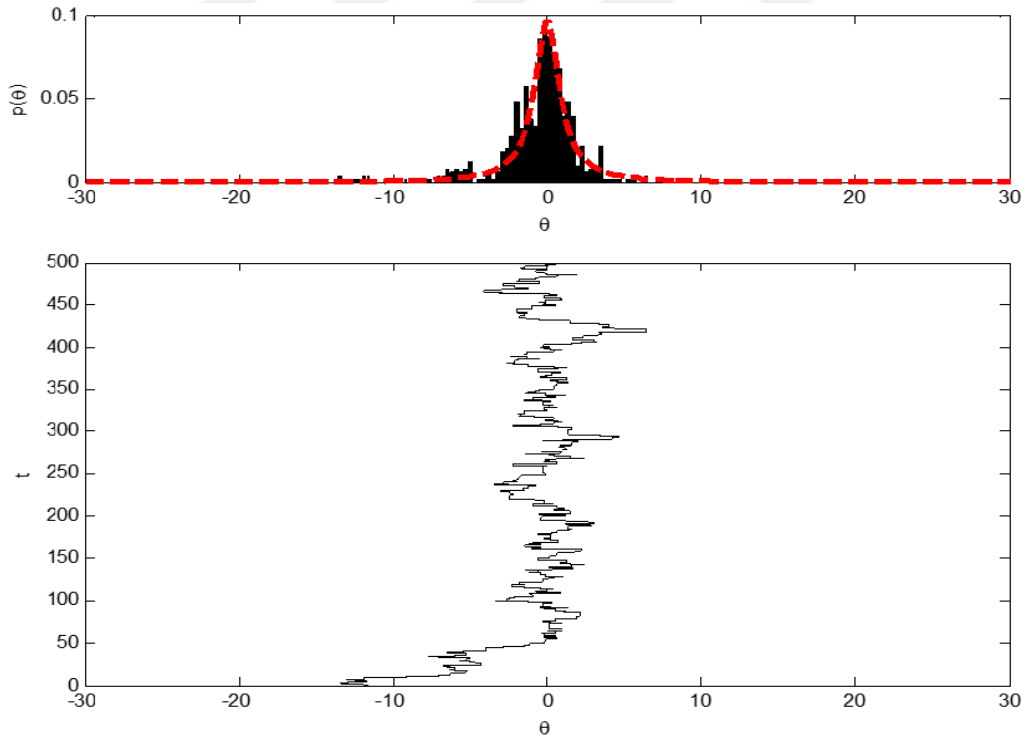
olmak üzere MCMC yöntemlerinden Metropolis algoritması kullanılarak bu dağılımdan sayı üretilmek istensin. Bir önceki bölümde anlatıldığı gibi, Metropolis örnekleme yapabilmek için normalleştirme sabitine ihtiyaç yoktur. Yani,

$$f(\theta) \propto \frac{1}{(1+\theta^2)} \quad (5.2)$$

biçiminde alınabilir. Metropolis algoritmasına göre kabul olasılığı,

$$\alpha = \min\left(1, \frac{\text{Cauchy}(\theta^*)}{\text{Cauchy}(\theta(t))}\right) = \min\left(1, \frac{1+(\theta^{(t)})^2}{1+(\theta^{(*)})^2}\right) \quad (5.3)$$

biçiminde hesaplanır. Cauchy dağılımından sayı üretmek için öneri dağılımı olarak değer aralığı Cauchy dağılımına benzeyen Normal dağılım alınabilir. Örneğin, standart sapması $\sigma = 1$ olarak bilinen $\theta^{(t)}$ ortalamalı Normal dağılım alınsın. $\theta^{(t)}$ için de başlangıç değeri olarak Düzgün dağılım alınabilir. Simülasyonda, başlangıç değeri $\theta^{(0)} \sim U(-30,30)$ olan dağılımdan rasgele üretilmiştir. Bir sonraki θ değeri ortalaması $\theta^{(0)}$, varyansı 1 olan Normal dağılımdan üretilip kabul olasılığı hesaplanmış ve bu şekilde devam edilerek $n = 500$ iterasyon için üretilen zincir aşağıdaki gibi elde edilmiştir.



Şekil 5.1 Metropolis Algoritması ile Cauchy dağılımından örnekleme

Grafikte, siyah çubuklar ile ifade edilen histogram Metropolis algoritmasıyla elde edilen örneklerin histogramıdır yani elde edilmek istenilen Cauchy olasılık yoğunluk

fonksiyonunun bir tahminidir. Histogramın üzerine çizilen kırmızı kesikli eğri ise teorik (normalize edilmemiş) Cauchy yoğunluk fonksiyonunu göstermektedir. Dikkat edilirse örneklem sonucunda elde edilen histogram ile elde edilmek istenilen oyf birbirine oldukça yakındır. Yani algoritma istenilen dağılımdan sayı üretme konusunda oldukça başarılıdır. İkinci grafikte ise üretilen θ değerlerine ilişkin iz grafiği verilmiştir. Bu grafik ile zincirin nasıl davrandığı gözlenmiş olur. Dikkat edilirse zincir bir süre sonra merkez noktası etrafında dolanmaya başlamıştır. Bazı bölgelerde (özellikle 0'ın altında ve 10'un üzerinde) ani sıçramalar olması Cauchy dağılımının kuyruklarının ağır olmasından dolayıdır.

5.2 Normal Dağılım Parametrelerinin Sonsal Dağılımlarından Gibbs Örnekleme Yöntemi İle Örneklem Alınması

$Y_1, Y_2, \dots, Y_n \sim N(\mu, \sigma^2)$ olan Normal dağılımdan bir örneklem olsun. μ ve σ^2 parametreleri bilinmiyor ve MCMC yöntemlerinden Gibbs örnekleme kullanılarak bu parametreler tahmin edilmek isteniyor olsun. Buna göre, Bayesci yaklaşım mantığıyla μ ve σ^2 parametreleri için sonsal dağılımlar elde edilmelidir. Bunun için ilk olarak parametrelere ilişkin önsel dağılımlar belirlenmelidir.

$$\begin{aligned}\mu &\sim N(\mu_0, \tau_0^2), \\ \sigma^2 &\sim \text{Inverse Gamma}(\alpha_0, \beta_0)\end{aligned}\tag{5.4}$$

olsun. Önsel dağılımlar belirlenip, gözlemler yapıldıktan sonra parametrelere ilişkin sonsal dağılımlar belirlenmelidir. Gibbs örneklemesini kullanabilmek için parametrelere ilişkin koşullu dağılımların bilinmesi gerekir. Gözlemler, $y, y_2, \dots, y_n$ olmak üzere sonsal dağılımlar,

$$\begin{aligned}\mu | \sigma^2, y &\sim N(\mu_n, \tau_n^2), \\ \tau_n^2 &= \left(\frac{n}{\sigma^2} + \frac{1}{\tau_0^2}\right)^{-1}, \mu_n = \tau_n^2 \left(\frac{n\bar{y}}{\sigma^2} + \frac{\mu_0}{\tau_0^2}\right)\end{aligned}\tag{5.5}$$

$$\sigma^2 | \mu, y \sim \text{Inverse Gamma}(\alpha_n, \beta_n)$$

$$\alpha_n = \alpha_0 + \frac{n}{2}, \beta_n = \beta_0 + \frac{1}{2} \sum_{i=1}^n (y_i - \mu)^2$$

biçiminde elde edilir (Gelman vd., 2013).

Simülasyon çalışmasında $\mu = 5, \sigma^2 = 4$ olan Normal dağılımdan $n = 100$ gözlem alınmıştır. Gibbs algoritması kullanarak parametrelere ilişkin sonsal dağılımlardan gözlem alabilmek için, ilk olarak $\mu_0 = 1$ ve $\sigma_0^2 = 1$ başlangıç değerleri alınmıştır. Eldeki veriler ve parametreler için alınan başlangıç değerleri kullanılarak ilk olarak μ ' nün koşullu dağılımından daha sonra buradan elde edilen μ değeri kullanılarak σ^2 ' nin koşullu dağılımından örnekler türetilir. Daha sonra bu işlemlere ardışık olarak üretilmek istenen örneklem sayısına ulaşılan kadar devam edilir. Simülasyon çalışmasında önsel dağılımlar,

$$\mu \sim N(0,1)$$

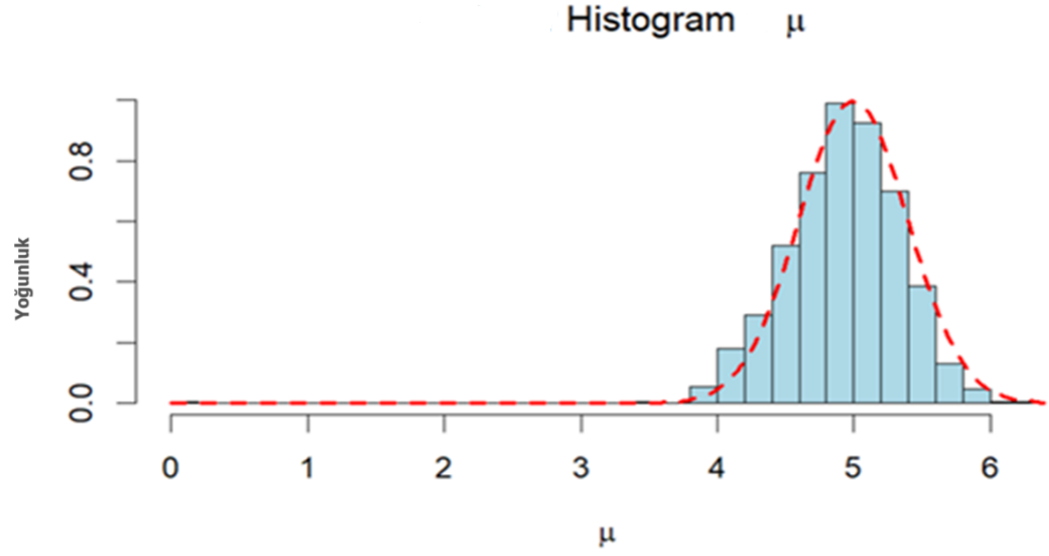
$$\sigma^2 \sim \text{Inverse Gamma}(1,1)$$

olarak alınmıştır. Buna göre, sonsal dağılımlar

$$\mu | \sigma^2, y \sim N\left(\frac{\sum_{i=1}^n y_i}{n}, \frac{\sigma^2}{n}\right)$$

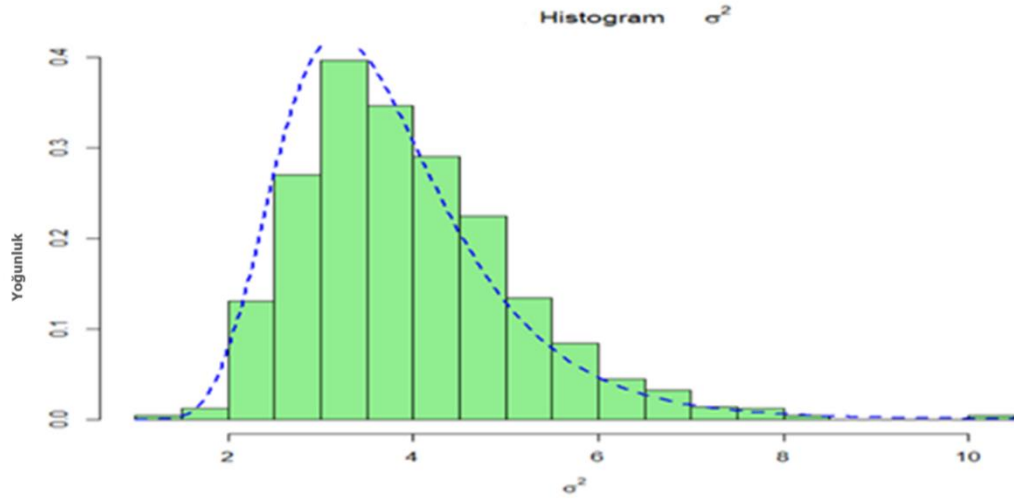
$$\sigma^2 | \mu, y \sim \text{Inverse Gamma}\left(\frac{n}{2}, \frac{\sum_{i=1}^n (y_i - \mu)^2}{2}\right) \quad 5.6$$

biçiminde elde edilmiştir.



Şekil 5.2 Gibbs Algoritması ile Normal Dağılımın μ parametresinin sonsal dağılımında örnekleme

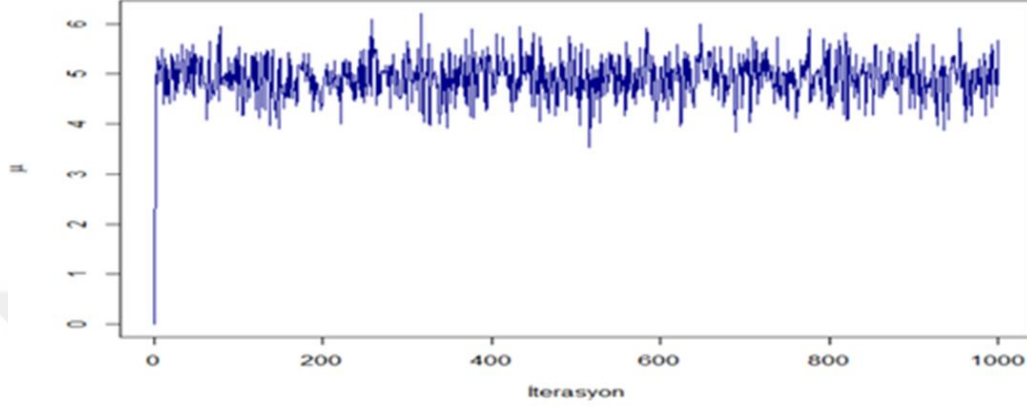
Histogram, Gibbs örneklemesiyle elde edilen μ örneklerinin dağılımını göstermektedir. Görüldüğü gibi dağılım çan eğrisi şeklinde, yani normal dağılıma yakın bir şekil almıştır.



Şekil 5.3 Gibbs Algoritması ile Normal Dağılımın σ^2 parametresinin sonsal dağılımından örnekleme

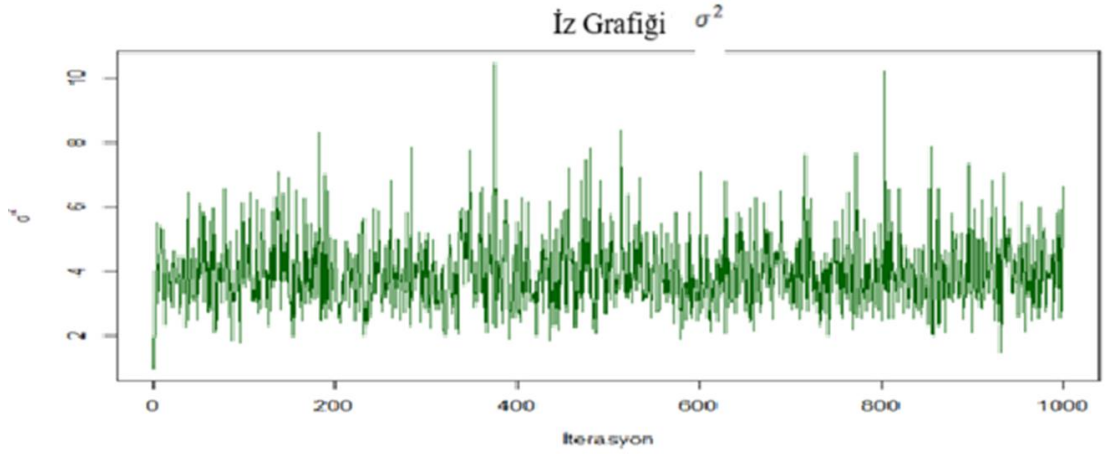
Şekil 5.3'te histogram ile teorik olarak elde edilen oyf eğrisinin üst üste geldiği görülmektedir. Buna göre, Gibbs örneklemesinin başarılı olduğu, örneklerin gerçekten de

teorik olarak elde edilen sonsal dağılımdan geldiği söylenebilir. Histogram, σ^2 için elde edilen örneklerin sağa çarpık olduğunu göstermektedir ki Inverse Gamma dağılımının yapısından bu beklenen bir durumdur.



Şekil 5.4 Normal Dağılımın μ parametresinin sonsal dağılımından elde edilen iz grafiği

Grafik, yaklaşık olarak 5 etrafında dalgalanma eğiliminde gözleniyor. Bu da, gerçek μ değeri olan 5 ile oldukça uyumludur. Bu, zincirin doğru hedef dağılıma yakın bir aralıkta örnekleme yaptığını gösterir. Başlangıç değeri $\mu = 0$ olduğundan, zincir ilk 20–30 iterasyonda hızla 5 civarına geçiş yapmıştır. Bu bölge, burn-in olarak kabul edilerek, istenilirse analize dahil edilmeyebilir. Zincir sürekli yukarı-aşağı hareket halinde gözükmektedir. Buna göre, iyi bir karışımın olduğu söylenebilir. Bu da örneklerin birbirinden bağımsız olmaya yakın olduğunu, yani örnekleme verimli olduğunu gösterir.



Şekil 5.5 Normal Dağılımın σ^2 parametresinin sonsal dağılımından elde edilen iz grafiği

σ^2 için elde edilen iz grafiğine bakıldığında grafikteki çizgi 4 etrafında yukarı-aşağı dalgalanma gösteriyor ve yaklaşık olarak 2 ile 6 arasında belirli bir aralıkta salınıyor. Buna göre zincirin iyi karıştığı ve hedef dağılım civarında dolaştığı söylenebilir. Grafikte, belli bir trend veya sistematik sapma gözlemlenmiyor. Buna göre de zincirin durağan hâle geldiği söylenebilir. Grafikte zaman zaman uç değerlere olduğu görülüyor bu da σ^2 'nin sonsal dağılımının kuyruklu olduğunu gösterir. Normal modelde de zaten σ^2 için kullanılan sonsal dağılım yani Inverse-Gamma dağılımı asimetrik ve kuyrukcludur. Bu nedenle, bu tarz sıçramalar, dağılımın doğasına uygundur ve modelin yanlış olduğunu göstermez.

6. DİNAMİK MODELLERDE MCMC YÖNTEMLERİ ÜZERİNE GERÇEK VERİ UYGULAMASI

Bu uygulamada Bitcoin (BTC) gibi yüksek volatiliteye sahip kripto para biriminin fiyat hareketleri üzerine dinamik model kurularak, zamanla değişen oynaklığı (volatilitayı) MCMC yöntemleri ile anlamaya ve tahmin etmeye çalışılmıştır. Uygulamanın temel amacı, finansal piyasalarda volatilitate tahmini çerçevesinde MCMC yöntemlerinin pratikteki işleyişini ve avantajlarını ortaya koymaktır.

Bu amaçla 01.01.2020-10.04.2025 tarih aralığında günlük BTC kapanış fiyatları (getiriler) Yahoo Finance web sayfası üzerinden elde edilmiş, literatür taramasıyla ve deneme yoluyla veri setine en uygun olduğu düşünülen Açıklayıcı Değişkenli Stokastik Volatilitate (SV) modeli kurulmuş, Hamiltonian Monte Carlo (HMC) yöntemi kullanılarak R programı ile parametre tahminleri ve gelecek volatilitate tahmini yapılmıştır.

6.1 Modelin Yapısı

Bitcoin (BTC), bireylerin doğrudan birbirlerine, banka veya başka bir aracının (merkezi otorite) onayına ihtiyaç duymadan, güvenli bir şekilde dijital para gönderip alabildiği elektronik nakit sistemi olarak tasarlanmıştır (Nakamoto, 2008). Merkeziyetsiz yapısı, 7/24 işlem görmesi ve sınırlı arzı nedeniyle geleneksel finansal araçlara kıyasla çok daha yüksek oynaklık (volatilitate) gösterir. Kısa vadeli ani fiyat sıçramaları sık görülür, makroekonomik açıklamalar, regülasyon haberleri, borsa teknik seviyeleri, arz kısıtı gibi birçok faktörden etkilenir.

Finansal piyasalarda volatilitate, fiyatların zaman içindeki dalgalanma büyüklüğünü ifade eder. Bir varlığın getirilerinin zaman içerisindeki standart sapması ile ölçülür. Yüksek volatilitate, fiyatlarda ani ve büyük değişimlerin yaşandığını, düşük volatilitate ise görece daha istikrarlı bir seyir izlendiğini gösterir. Bu tür oynaklıklar, yatırımcı davranışlarını, piyasa likiditesini ve risk yönetimini doğrudan etkiler. Bu nedenle volatilitenin doğru şekilde modellenmesi önemlidir. Ancak volatilitate çoğu zaman doğrudan gözlemlenemez; bu nedenle gizli bir süreç olarak ele alınır. Stokastik Volatilitate (SV) modelleri bu

yaklaşımı benimseyerek, volatilitiyi zamanla değişen bir gizli değişken olarak tanımlar ve bu değişkenin hareketini ayrıca bir stokastik süreçle modellemenin yolunu sunar (Kim vd., 1998).

Bu çalışmada kullanılan, B. Kachnowski'nin (2020) önerdiği SV modelinden esinlenerek geliştirilen modelde, volatilitenin sadece geçmiş fiyat hareketlerinden değil, aynı zamanda dışsal faktörlerden (şok) de etkilendiği varsayılmıştır. Bu nedenle modele, ani ekonomik ve politik gelişmeleri temsil eden açıklayıcı değişkenler (regresörler) entegre edilmiştir. Bu şekilde volatilitenin tahmini sadece içsel dinamiklere değil, aynı zamanda dışsal faktörlere bağlı olarak da şekillenmiştir

Model 01.01.2020 – 01.05.2025 tarihleri arasındaki günlük BTC kapanış fiyatlarından hesaplanan logaritmik getiriler kullanılarak tahmin edilmiştir. Bu dönemde kripto para piyasasında COVID-19 kaynaklı çöküşler, 2021 boğa piyasası (yükseliş dönemi), Terra krizi gibi ciddi oynaklık dönemleri yaşanmıştır. Bu yapılar, volatilitenin değişken doğasını modellemek için uygun bir test alanı sağlamaktadır.

Finansal zaman serilerinde, fiyat serisinin yerine logaritmik getiri serisi kullanılması yaygın bir uygulamadır. Bu dönüşüm, hem verinin daha istatistiksel olarak simetrik hale gelmesini sağlar hem de volatilitenin modellemeleri için uygun bir yapı sunar (Tsay, 2005). Logaritmik getiri serisi şu şekilde tanımlanır:

$$y_t = \log(P_t) - \log(P_{t-1}) \quad (6.1)$$

Burada y_t , t dönemine ait logaritmik getiri, P_t , t zamanındaki kapanış fiyatıdır. Bu dönüşüm ile elde edilen getiri serisi, daha simetrik bir yapıya kavuşur ve SV modellerinde kullanılmaya uygun hale gelir (Kim, Shephard ve Chib, 1998).

Getiri serisi şu şekilde modellenir:

$$y_t = \mu_y + \sigma_t \cdot \varepsilon_t \quad , \quad \varepsilon_t \sim N(0,1) \quad (6.2)$$

Burada μ : ortalama getiri çoğu uygulamada sıfır kabul edilir. SV modellerinde zamanla değişen volatilitiyi doğrudan modellemek yerine, onun logaritması kullanılarak logaritmik varyans tanımlanır

$$\log(\sigma_t^2) = h_t \quad \Rightarrow \quad \sigma_t = e^{h_t/2} \quad (6.3)$$

Bu dönüşüm, volatilitenin pozitif olmasını sağlar. Bu durumda ortalama getiri $\mu_y = 0$ varsayımıyla model şu hale gelir:

$$y_t = e^{h_t/2} \cdot \varepsilon_t \quad , \quad y_t \sim N(0, e^{h_t}) \quad (6.4)$$

Buradaki h_t , gözlemlenemeyen log-volatilite sürecidir. SV modellerinde bu süreç genellikle bir AR(1) yapıda modellenir. Bu çalışmada kullanılan model, klasik SV yapısına ek olarak, geçmiş dönemde yaşanan dışsal şokların (örneğin ekonomik kriz, savaş, politika değişiklikleri vb.) etkisini açıklayıcı değişken vektörü aracılığıyla hesaba katar. Modelin yapısı şu şekildedir:

$$h_t = \mu + \phi(h_{t-1} - \mu) + \gamma X_t^T + \tau \eta_t \quad (6.5)$$

Burada:

- y_t : log-getiri
- h_t : log-volatilite (gizli)
- μ : ortalama volatilite düzeyi,
- ϕ : volatilite sürecindeki otokorelasyon katsayısı,
- γ : açıklayıcı değişkenlerin volatilite üzerindeki etkisini belirleyen katsayılar vektörü,
- X_t : t anındaki şokları ve dışsal faktörleri temsil eden açıklayıcı değişken vektörü
- τ : log-volatilite sürecinin hata teriminin standart sapması,

- η_t : hata terimi ($\eta_t \sim N(0,1)$)

Modelde kullanılan X_t açıklayıcı değişken vektörü aşağıdaki dışsal faktörleri-şokları kapsamaktadır:

- Ekonomik şoklar: faiz ve enflasyon kararları
- Politik şoklar: savaş ve yaptırım haberleri
- Kripto para piyasası krizleri: FTX, Celsius gibi kripto para kaynaklı finansal krizler
- Teknik olaylar: Bitcoin halving, Taproot güncelleme gibi büyük ağ olayları
- Sert fiyat düşüşleri: BTC'nin günlük getirilerinin %10'dan fazla düştüğü günler,

Bu açıklayıcı değişkenler 0–1 biçiminde kodlanmış ikili gösterge değişkenler olarak modele entegre edilmiştir ve yalnızca geçmiş gözlemler üzerinden modele dahil edilmiştir.

Yukarıdaki yapı sayesinde hem volatilitenin zamansal bağımlılığı hem de açıklayıcı faktörlerin etkisi aynı anda modellenmiş olur. Modelin tamamı, açıklayıcı değişken içeren bir stokastik volatilité modeli olarak tanımlanabilir.

6.2 HMC Algoritması ile Günlük Bitcoin Volatilitésinin Modellenmesi

Kripto para piyasaları, yapısal rejim değişimleri, yüksek volatilité ve dışsal şoklara duyarlılığı nedeniyle klasik istatistiksel modellerle tahmin edilmesi güç sistemlerdir. Bitcoin fiyatları zaman içerisinde boğa (yükseliş), ayı (düşüş) ve yatay (durağan) piyasa rejimleri arasında geçişler yapabilir. Bu da getirilerin çok modlu ve karmaşık bir dağılım yapısı sergilemesine neden olur. Bu tür dağılımlar genellikle normalize edilemeyen ve kapalı formda gösterilemeyen karmaşık yapılara sahiptir, dolayısıyla geleneksel istatistiksel yöntemlerle doğrudan modellenmesi zordur (Robert ve Casella, 2004).

Bu çalışmada, Bitcoin log-getiri serisinin volatilité süreci, modelin Yapısı bölümünde açıklanan açıklayıcı değişkenli stokastik volatilité modeli çerçevesinde modellenmiş ve

HMC algoritmasına dayalı No-U-Turn Sampler (NUTS) yöntemi kullanılarak parametre tahmini gerçekleştirilmiştir.

HMC algoritması, parametre uzayında daha uzun ve sistematik sıçramalar yaparak, yüksek otokorelasyon problemini azaltan ve hedef dağılımı daha etkin keşfeden bir örnekleme yöntemidir. NUTS ise HMC algoritmasındaki adım sayısının manuel olarak belirlenmesi sorununu ortadan kaldırmak amacıyla geliştirilmiş bir eklentidir. NUTS, simülasyon sırasında zincirin geri dönme eğilimi gösterdiği anı tespit ederek örnekleme sürecini otomatik olarak durdurur. Böylece hem verimlilik artırılır hem de hesaplama maliyeti optimize edilir.

Bu çalışmada HMC-NUTS algoritması, rstan paketi kullanılarak R programında uygulanmış ve dört bağımsız zincir üzerinden 4000 iterasyon çalıştırılmıştır. Her zincir için 1000 iterasyon “burn-in” (ısınma) dönemi olarak ayrılmış, toplamda 12.000 kullanılabilir örnek elde edilmiştir.

MCMC çıktılarının ne kadar doğru-hatalı olabileceğini ölçmek, modelleme sürecinin bir parçasıdır. Gelman-Rubin $\hat{R}$ değeri, MCMC zincirlerinin hedef dağılıma yakınsamış olup olmadığını değerlendirir. Zincirler aynı hedef dağılıma yakınsadığında, $\hat{R}$ değeri 1'e çok yakın olmalıdır. Aynı modelle birkaç farklı başlangıç noktasından MCMC zinciri çalıştırılır (Gelman vd., 2013). Her zincir için iç varyans (within-chain variance) ve zincirler arası varyans (between-chain variance) hesaplanır. $\hat{R}$, bu varyansların oranına dayanarak hesaplanır:

$$\hat{R} = \sqrt{\frac{\hat{V}}{W}} \quad (6.6)$$

- $\hat{V}$: zincir içi varyansların ortalaması
- W : toplam (zincirler arası + zincir içi) varyansın tahmini

Çizelge 6.1 Gelman-Rubin $\hat{R}$ değerleri anlam tablosu

$\hat{R} \approx 1$	Zincirler yakınsamış; sonuçlar güvenilirdir.
$\hat{R} > 1.1$	Zincirler yakınsamamış; daha fazla iterasyona ihtiyaç var veya zincirler karışmıyor.
$\hat{R} > 1.2$	Ciddi yakınsama problemi var. Model, örnekleyici veya başlangıçlar gözden geçirilmeli

MCMC uygulamalarında karşılaşılan temel sorunlardan biri, zincir içerisindeki örneklerin birbirine bağımlı olmasıdır. Bu bağımlılık, otokorelasyon yoluyla örneklerin taşıdığı bilginin etkinliğini azaltır. Bu nedenle, etkin örnek büyüklüğü (n_{eff}) kavramı geliştirilmiştir. Etkin örnek büyüklüğü, zincirdeki toplam iterasyon sayısı (N) ile otokorelasyon katsayıları (ρ_k) üzerinden aşağıdaki şekilde hesaplanır:

$$n_{eff} = \frac{N}{1 + 2 \sum_{t=1}^{\infty} \rho_t} \quad (6.7)$$

HMC algoritmasında her iterasyonda güncellenen parametrelerin bilgi değeri, bu otokorelasyon yapısına göre değerlendirilir. Yüksek n_{eff} değerleri, zincirin iyi karıştığını ve tahminlerin istatistiksel olarak güvenilir olduğunu gösterir (Stan Development Team, 2019).

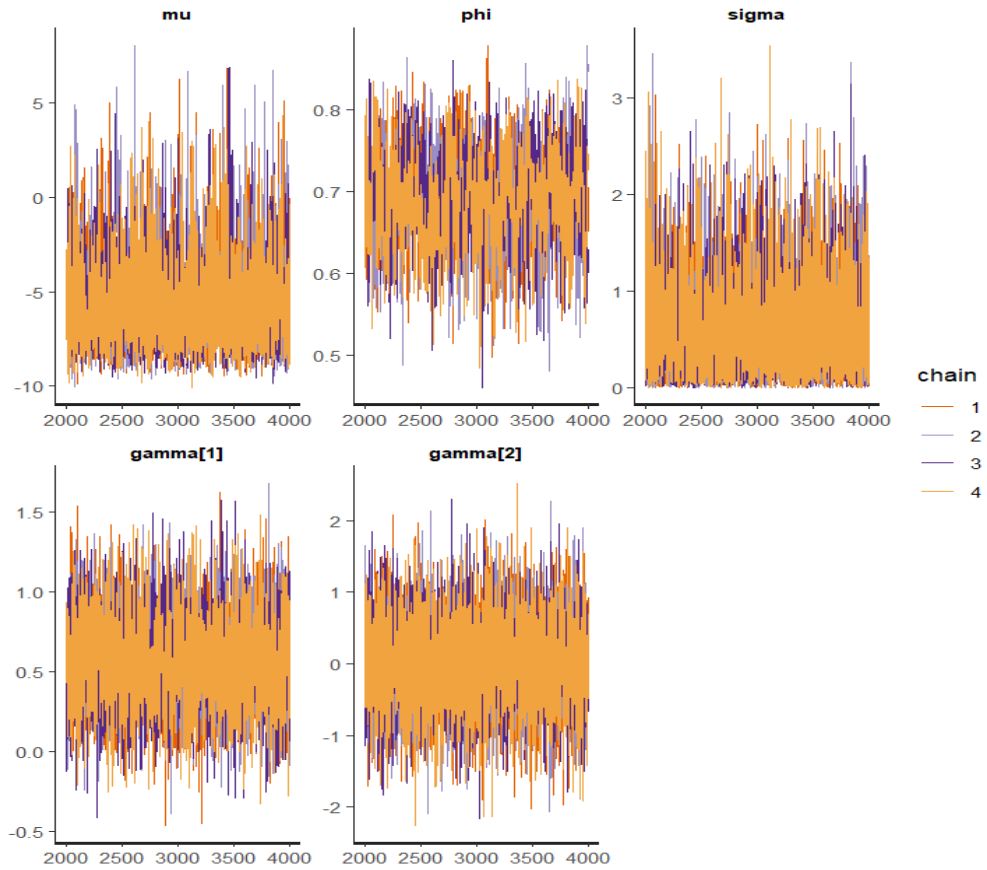
Model parametrelerine ilişkin MCMC çıktıları Tablo 6.2’de özetlenmiştir.

Çizelge 6.2 $\hat{R}$ (Gelman-Rubin $\hat{R}$) ve n_{eff} değerleri

Parametre	Açıklama	$\hat{R}$ Değeri	n_{eff}
μ	Ortalama log-volatilite düzeyi	1.0017	2917
ϕ	Log-volatilite sürecindeki otokorelasyon katsayısı	1.0043	692
σ	Volatilite inovasyonlarının standart sapması	1.0001	6905
τ	Dışsal şokların volatilite üzerindeki etkisinin ölçek parametresi	1.0044	705
γ_1	Açıklayıcı değişken BTC fiyatında %10’dan fazla düşüş yaşanan günlerin etkisi	0.9996	4073
γ_2	Makroekonomik şokların etkisi (faiz kararı, TÜFE, COVID-19 vb.)	1.0004	6537
γ_3	Kripto kaynaklı finansal krizlerin etkisi (FTX, Celsius vb.)	1.0012	2693
γ_4	Teknik olayların etkisi (halving, Taproot vb.)	0.9997	6841
γ_5	Politik şokların etkisi (savaşlar, düzenleyici müdahaleler)	1.0001	7476

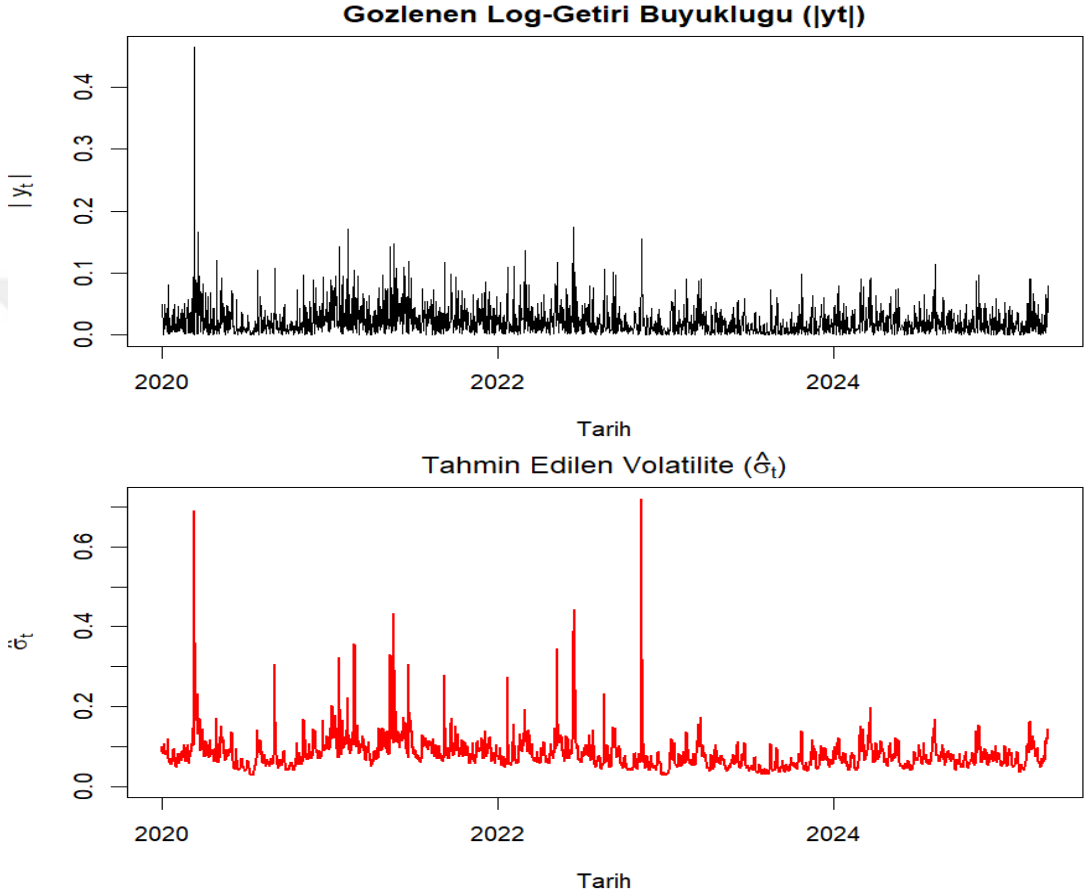
Tablo 6.2’de oluşturulan modele ait parametrelerin Gelman-Rubin $\hat{R}$ istatistikleri sunulmuştur. Buna göre tüm parametreler için $\hat{R}$ değeri 1.01’in altındadır. Bu, örnekleme zincirlerinin yeterince karıştığını ve sonsal tahminlerin güvenilir olduğunu göstermektedir.

Modeldeki ana parametrelerin çoğunda n_{eff} değeri 2000’in üzerindedir; bu da örneklerin yeterince bağımsız olduğunu ve modelin iyi karıştığını göstermektedir. Özellikle volatilité parametreleri için n_{eff} değerlerinin 6000–7000 bandında olması, tahminlerin güvenilirliğini artırmaktadır. Öte yandan ϕ ve τ gibi birkaç parametrede n_{eff} nispeten daha düşüktür (yaklaşık 700 civarında), bu da bu parametrelerde örnekler arasında daha yüksek otokorelasyon olabileceğini göstermektedir. Ancak bu değerler de kabul edilebilir düzeydedir.



Şekil 6.1 Dört HMC zinciri için iz grafikleri (traceplot)

Şekil 6.1’de, modeldeki ana parametreler için dört farklı HMC zincirine ait iz çizimleri (traceplot) sunulmuştur. Tüm zincirler hedef dağılım üzerinde kararlı biçimde salınım göstermekte olup, zincirler arasında herhangi bir sapma veya ayrışma gözlenmemektedir. Bu durum, örnekleme sürecinin başarılı bir yakınsama sağladığını göstermektedir.



Şekil 6.2 Gerçekleşen log-getiri mutlak değerleri ($|y_t|$) ve tahmin edilen volatiliteler ($\hat{\sigma}_t$)

Şekil 6.2’de model tarafından tahmin edilen günlük volatiliteler, yüksek oynaklık dönemlerinde $|y_t|$ büyüklükleriyle uyumlu şekilde hareket etmektedir. Özellikle şokların yaşandığı tarihlerde (örneğin Terra çöküşü, faiz kararları), tahmin edilen volatiliteler belirgin artış göstermiştir.

Modelin tahmin başarısını sadece grafiksel olarak değil, aynı zamanda sayısal bir hata metriğiyle de değerlendirmek amacıyla, tahmin edilen log-volatiliteler ile gerçek log-volatiliteler karşılaştırılmıştır. Bu karşılaştırmada, tahmin edilen değerler $\hat{h}_t$

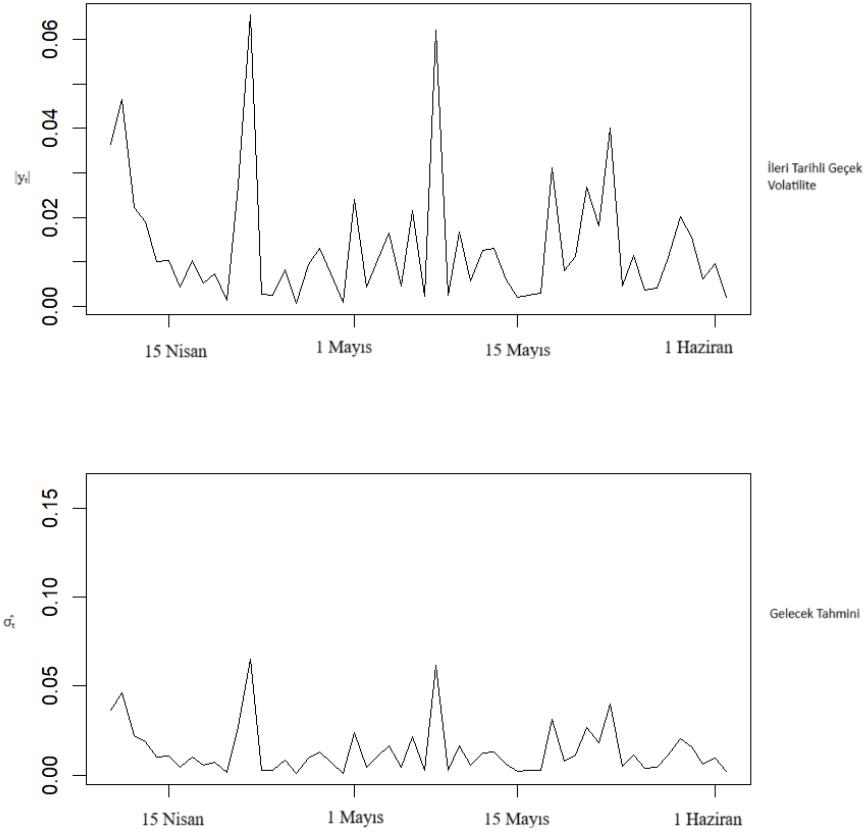
ile gerçekleşen değerler $\log(|y_t|)$ arasındaki farklar kullanılarak Ortalama Kare Hata (MSE) ve Ortalama Mutlak Hata (MAE) hesaplanmıştır.

Bu metrikler sırasıyla şu şekilde tanımlanır:

$$MSE = \frac{1}{T} \sum_{t=1}^T (\hat{h}_t - \log(|y_t|))^2$$
$$MAE = \frac{1}{T} \sum_{t=1}^T (|\hat{h}_t - \log(|y_t|)|) \quad (6.8)$$

Bu değerler, modelin ne derece başarılı tahminler yaptığını değerlendirmek için sıklıkla kullanılan standart hata ölçütleridir (Hyndman ve Koehler, 2006).

Uygulama sonucunda elde edilen değerler $MSE = 1.666$, $MAE = 1.126$ olarak bulunmuştur. Bu sonuçlar, modelin log-volatilite tahminlerinde belirli bir doğruluk seviyesine ulaştığını göstermektedir.



Şekil 6.3 10.04.2025-01.06.2025 arası tahmin edilen volatilite ve gerçekleşen getiriler

Şekil 6.3 ile 10.04.2025 – 01.06.2025 geleceğe yönelik yapılan kısa vadeli volatilite tahminleri ile gözlenen log-getirilerin görsel olarak karşılaştırılması verilmiştir. Model, hem artış hem düşüş yönündeki oynaklık sinyallerini başarılı şekilde öngörmektedir.

Bu uygulama, HMC yönteminin Bitcoin gibi yüksek oynaklık içeren sistemlerde etkin bir örnekleme aracı olduğunu göstermektedir. Özellikle çok modlu yapılar ve dışsal şoklara dayalı açıklayıcı değişkenlerin dahil edildiği modellerde HMC'nin hızlı yakınsama sağlaması ve yüksek etkin örnek üretmesi önemli bir avantajdır. Açıklayıcı değişkenleri çoğunun volatilite üzerinde istatistiksel olarak anlamlı etki göstermesi, modelin açıklayıcılığını artırmaktadır.

7. SONUÇ

Bu tez çalışmasında, MCMC yöntemlerinin hem teorik temelleri hem de dinamik sistemlerdeki uygulamaları ayrıntılı biçimde ele alınmıştır. Tezin teorik kısmında Bayesci istatistik, MCMC simülasyonu kavramları açıklanmış; MCMC yöntemleri, özellikle Metropolis-Hastings ve HMC algoritmaları detaylandırılmıştır.

Uygulama kısmında ise, dinamik bir finansal süreç olan Bitcoin volatilitesi, açıklayıcı değişkenli SV modeli ile modellenmiştir. Bu modelde, volatilité gizli bir süreç olarak tanımlanmış ve hem geçmiş fiyat hareketleri hem de dışsal şoklar dikkate alınmıştır. Modelin tahminleri, HMC algoritması ve No-U-Turn Sampler (NUTS) yardımıyla gerçekleştirilmiştir.

HMC ile tahmin edilen parametrelerin tümü için $\hat{R}$ istatistikleri 1.01'in altındadır. Bu da MCMC zincirinin hedef dağılıma yakınsadığını göstermektedir. Etkin örnek büyüklükleri (n_{eff}), çoğu parametre için 2000'in üzerinde gerçekleşmiş, bu da zincirin karışım kalitesinin yüksek olduğunu ve tahminlerin güvenilirliğini göstermektedir.

Tahmin edilen volatilité değerleri, yüksek oynaklık dönemlerinde gerçekleşen log-getirilerin mutlak değerleriyle paralel hareket etmiştir. Bu paralelliği daha sistematik bir biçimde değerlendirmek için tahmin edilen log-volatilité ile gerçekleşen log-getiri mutlak değerleri arasındaki ortalama kare hata (MSE) ve ortalama mutlak hata (MAE) hesaplanmıştır. Bu değerler sırasıyla $MSE = 1.666$, $MAE = 1.126$ olarak bulunmuştur. Görsel analizler (Şekil 2 ve Şekil 3) tahmin edilen volatilitenin özellikle makroekonomik şoklar ve kripto krizleri sırasında güçlü bir öngörü performansı sergilediğini göstermektedir.

Çalışmada yalnızca 2020-2025 dönemine ait BTC fiyatları ve manuel tanımlanan şoklar kullanılmıştır. Otomatik şok tespiti veya alternatif veri kaynaklarıyla modelin kapsayıcılığı artırılabilir.

Volatilite süreci için sabit parametrelili bir AR(1) yapısı ve normal dağılımlı hata terimi varsayılmıştır. Ancak finansal zaman serilerinde sıklıkla gözlenen özellikler, örneğin değişen varyans, aşırı uç değerler ve volatilitenin zaman içinde değişen yapısı, bu varsayımlarla tam olarak yakalanamayabilir. Bu nedenle, gelecekteki çalışmalarda t-dağılımlı hata terimleri, zamanla değişen parametrelili yapılar veya volatilitenin kendi iç varyansını da modelleyen volatilitenin volatilitesi türü yaklaşımlarla modelin çıktıları karşılaştırılabilir. Bu tür genelleştirmeler, modelin özellikle uç senaryolarda daha sağlam tahminler üretmesine katkı sağlayabilir.

Ayrıca açıklayıcı değişken olarak kullanılan şok vektörleri manuel olarak oluşturulmuş ve yalnızca geçmişte gerçekleşen belirli olaylara dayandırılmıştır. Bu durum hem kapsam hem de nesnellik açısından sınırlayıcı olabilir. Gelecek araştırmalarda, daha sistematik şok tespiti yöntemlerinin (örneğin doğal dil işleme ile haber analizi veya sosyal medya verisi kullanımı) kullanılması modelin genişletilmesine katkı sağlayabilir.

Son olarak, Hamiltonian Monte Carlo (HMC) algoritmasının hesaplama yükü nedeniyle daha uzun örnekleme zincirleri çalıştırılmamış, bazı parametrelerde etkin örnek büyüklüğü sınırlı kalmıştır. Alternatif örnekleme algoritmaları veya paralel hesaplama teknikleri kullanılarak bu sınırlama aşılabılır.

7.1 Gelecek Çalışmalar İçin Öneriler

Gelecek çalışmalara yönelik potansiyel araştırma alanları aşağıda sıralanmıştır:

- Açıklayıcı değişken belirleme sürecinde otomasyon: Geçmişteki şokların etkisini tespit etmek amacıyla kullanılan açıklayıcı değişkenlerin daha nesnel ve sistematik biçimde oluşturulabilmesi için, doğal dil işleme yöntemleri veya haber/sosyal medya analizi gibi otomatik bilgi çıkarım tekniklerinin entegrasyonu önerilmektedir.
- Farklı piyasa araçlarına uygulama: Modelin, Bitcoin dışındaki kripto varlıklar veya geleneksel finansal enstrümanlar üzerinde test edilerek genellenebilirliğinin sınanması, literatüre katkı sağlayacaktır.

- Gerçek zamanlı tahmin sistemlerinin geliştirilmesi: Model çıktılarının operasyonel süreçlerde kullanılabilmesi adına, gerçek zamanlı veri akışıyla entegre çalışan tahmin sistemlerinin geliştirilmesi önem arz etmektedir.
- Risk ölçüm modelleri ile bütünleşme: Elde edilen volatilité tahminlerinin Value-at-Risk (VaR) gibi risk tahmini modelleriyle entegre edilmesi, finansal risk yönetimi uygulamaları açısından pratik fayda sunabilir.



KAYNAKLAR

- Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. arXiv preprint arXiv:1701.02434.
- Bronk Ramsey, C. (2009). Bayesian analysis of radiocarbon dates. *Radiocarbon*, 51(1), 337–360. <https://doi.org/10.1017/S0033822200033865>
- Brooks, S., Gelman, A., Jones, G. L., ve Meng, X. L. (2011). *Handbook of Markov Chain Monte Carlo*. CRC Press, Boca Raton.
- Cengiz, M. A., Terzi, E., Şenel, T. ve Murat, N. (2012). Lojistik regresyonda parametre tahmininde bayesci bir yaklaşım. *Afyon Kocatepe Üniversitesi Fen Bilimleri Dergisi*, 12(011302), 15–22
- Christensen, N., Meyer, R., Knox, L., ve Luey, B. (2001). Bayesian methods for cosmological parameter estimation from cosmic microwave background measurements. *The Astrophysical Journal*, 556(2), 93–108. <https://doi.org/10.1086/321606>
- Çelik, İ. (2013). Markov zincirlerinin temel özellikleri ve çeşitli uygulamaları. Yüksek lisans tezi (basılmamış). Ordu Üniversitesi, Fen Bilimleri Enstitüsü.
- Demirci, M. (2016). Bayes teoremi ve teoremin işletme bölümünde uygulamaları. *The Journal of Academic Social Science Studies (JASSS)*, (43), 439–462. <https://doi.org/10.9761/JASSS3173>
- Earl, D. J., ve Deem, M. W. (2005). Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23), 3910–3916.
- Ekinci, O. (2019). *MCMC ve uygulamaları Markov zinciri Monte Carlo simülasyonu R uygulamaları ile MCMC*, Türkmen Kitabevi, İstanbul
- Gamerman, D., ve Lopes, H. F. (2006). *Markov chain Monte Carlo: Stochastic simulation for bayesian inference*. CRC Press, Boca Raton.
- Gelfand, A. E., ve Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410), 398–409. <https://doi.org/10.1080/01621459.1990.10476213>
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., ve Rubin, D. B. (2013). *Bayesian data analysis* (3. baskı). CRC Press, Boca Raton. Bölüm A.1.1, s. 587.
- Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood. *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface* (ss. 156–163). Interface Foundation.

- Griffiths, T. L., Kemp, C., ve Tenenbaum, J. B. (2008). Bayesian models of cognition. R. Sun (Ed.), *The Cambridge handbook of computational psychology* (pp. 59–100). Cambridge University Press.
https://cocosci.princeton.edu/tom/papers/bayeschapter.pdf?utm_source
- Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications". *Biometrika*, 57(1), 97-109.
- Hayes, Brian. (2023). First link in the Markov chain. *American Scientist*, 101(2), 92-97.
- Hoffman, M. D., ve Gelman, A. (2014). The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1), 1593–1623.
- Hyndman, R. J., ve Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679–688.
- Kachnowski, B. (2020). An empirical stochastic volatility model for Bitcoin price range Forecasting. *SSRN Electronic Journal*. <https://ssrn.com/abstract=3693387>
- Kastner, G., ve Frühwirth-Schnatter, S. (2014). Ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC estimation of stochastic volatility models. *Computational Statistics & Data Analysis*, 76, 408–423.
<https://doi.org/10.1016/j.csda.2013.01.002>
- Kim, S., Shephard, N., ve Chib, S. (1998). Stochastic volatility: Likelihood inference and comparison with ARCH models. *The Review of Economic Studies*, 65(3), 361–393.
- Kroese, D. P., Brereton, T., Taimre, T., ve Botev, Z. I. (2014). Why the Monte Carlo method is so important today. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6), 386–392. <https://doi.org/10.1002/wics.1314>
- Markov, A. A. (1906). Extension of the law of large numbers to dependent quantities. *Izvestiya Fiziko-Matematicheskogo Obshchestva pri Kazanskoy Universitete*, 15(2), 135–156.
- McGrayne, S. B. (2011). *The theory that would not die*, Yale University Press, New Haven.
- Metropolis, N. (1987). The beginning of the Monte Carlo method. *Los Alamos Scientific Special Issue*, 125–130.
- Metropolis, N., ve Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, 44(247), 335-341.
- Öztürk, F., ve Özbek, L. (2015). *Matematiksel modelleme ve simülasyon*. Pigeon Yayıncılık, Ankara.

- Planck Collaboration. (2018). Planck 2018 results, VI. Cosmological parameters. *Astronomy & Astrophysics*, 641, A6. <https://doi.org/10.1051/0004-6361/201833910>
- Rannala, B., ve Yang, Z. (1996). Probability distribution of molecular evolutionary trees: A new method of phylogenetic inference. *Journal of Molecular Evolution*, 43(3), 304–311. <https://doi.org/10.1007/PL00006090>
- Robert, C. P., ve Casella, G. (2004). *Monte Carlo statistical methods*. Springer, New York.
- Seneta, E. (2006). *Markov and the creation of Markov chains*. University of Sydney, Department of Statistics.
- Sims, C. A., ve Zha, T. (1998). Bayesian methods for dynamic multivariate models. *International Economic Review*, 39(4), 949–968. <https://doi.org/10.2307/2527347>
- Stan Development Team. (2019). *Stan reference manual* (Version 2.19). https://mc-stan.org/docs/2_19/reference-manual/effective-sample-size-section.html. Eriřim Tarihi: 01.06.2025
- Tsay, R. S. (2005). *Analysis of financial time series* (2nd ed.). John Wiley & Sons, Hoboken.
- Wilkinson, D. (2013). Parallel tempering and Metropolis coupled MCMC. <https://darrenjw.wordpress.com/2013/09/29/parallel-tempering-and-metropolis-coupled-mcmc/>. Eriřim Tarihi: 01.06.2025
- WMAP Science Team. (2011). Seven-year Wilkinson Microwave Anisotropy Probe (WMAP) observations: Cosmological interpretation. *The Astrophysical Journal Supplement Series*, 192(2), 18. <https://doi.org/10.1088/0067-0049/192/2/18>