

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И
ИНФОРМАТИКИ
Кафедра теории вероятностей и математической статистики

Ли Хуэй Ян

**Вычисление характеристик эффективности последовательных статистических
решающих правил**

Магистерская диссертация

Научный руководитель
Харин А.Ю., профессор, д.ф.-м.н.

Допущена к защите
«_____» _____ 2025 г.
Заведующий кафедрой _____

Минск, 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	1
ГЛАВА 1	2
ВВЕДЕНИЕ И ОБЗОР ИССЛЕДОВАНИЯ	2
1.1 Предыстория и значимость исследования	2
1.2 Текущее состояние исследований в стране и за рубежом	3
1.3 Содержание исследования и инновация	4
ГЛАВА 2	5
СВЯЗАННЫЕ ОСНОВНЫЕ ТЕОРИИ	5
2.1 Базовая теория цепей Маркова	5
2.2 Последовательная статистическая теория принятия решений	6
2.3 Метод расчета характеристик эффективности	8
2.4 Обзор методов аппроксимации	11
ГЛАВА 3	13
ПОСТРОЕНИЕ МОДЕЛИ, АППРОКСИМИРУЮЩЕЙ ЦЕПЬ МАРКОВА ...	13
3.1 Введение в проблему и основные предположения	13
3.2 Автокорреляционный анализ и марковская проверка	14
3.3 Метод динамической квантильной дискретизации	15
3.4 Снижение размерности главного компонентного анализа	16
3.5 Оценка вероятностей перехода	18
3.6 Проверка и анализ модели	18
ГЛАВА 4	20
ПОСЛЕДОВАТЕЛЬНАЯ СТАТИСТИЧЕСКАЯ МОДЕЛЬ ПРИНЯТИЯ	
РЕШЕНИЙ НА ОСНОВЕ АППРОКСИМИРУЮЩЕЙ ЦЕПИ МАРКОВА	20
4.1 Построение модели последовательного статистического решения	20
4.2 Алгоритм прогнозирования оставшегося срока службы	20
4.3 Метод расчета эксплуатационных характеристик	21
4.4 Оптимизация фронта Парето	22
4.5 Механизм самоадаптации модели	23
ГЛАВА 5	24
ЭКСПЕРИМЕНТ И АНАЛИЗ РЕЗУЛЬТАТОВ	24
5.1 Экспериментальный дизайн	24
5.2 Анализ чувствительности параметров	25
5.3 Основные экспериментальные результаты	28
5.4 Результаты и обсуждение	30
5.5 Анализ фронта Парето	30
5.6 Тест статистической значимости	31
ГЛАВА 6	33
ЗАКЛЮЧЕНИЕ И ПЕРСПЕКТИВЫ	33
6.1 Основные результаты исследований и инновации	33
6.2 Ограничения исследования и перспективы применения	33
ЗАКЛЮЧЕНИЕ	35
СПИСОК ЦИТИРУЕМОЙ ЛИТЕРАТУРЫ	36
ПРИЛОЖЕНИЯ	40

Реферат

Магистерская диссертация, 41 страниц, 4 рисунков, 50 источников, 61 формула

Ключевые слова: ПРИБЛИЖЕННАЯ ЦЕПЬ МАРКОВА, ПОСЛЕДОВАТЕЛЬНЫЙ СТАТИСТИЧЕСКИЙ РЕШАЮЩИЙ ПРАВИЛ, ВЫЧИСЛЕНИЕ ХАРАКТЕРИСТИК ЭФФЕКТИВНОСТИ, ОСТАВШИЙСЯ СРОК ПОЛЕЗНОГО ИСПОЛЬЗОВАНИЯ, ДИНАМИЧЕСКАЯ КВАНТИЛЬНАЯ ДИСКРЕТИЗАЦИЯ.

Объектом исследования являются приближенные марковские цепи и методы последовательного статистического принятия решений в задачах прогнозирования остаточного ресурса оборудования.

Предметом исследования является разработка и верификация вычислительного метода оценки характеристик производительности на основе приближенных марковских цепей для систем последовательного статистического принятия решений в задачах прогнозирования остаточного ресурса промышленного оборудования.

Целью работы является разработка вычислительно эффективного метода оценки характеристик производительности на основе аппроксимированных марковских цепей для систем мониторинга промышленного оборудования.

В ходе работы был разработан метод приближенной цепи Маркова и его применение в последовательном статистическом принятии решений для задачи прогнозирования остаточного ресурса промышленного оборудования. Предложена вычислительная модель, основанная на динамической квантильной дискретизации и снижении размерности РСА. Результаты показывают, что адаптивное обновление порога выборки каждые 100 циклов может уменьшить ошибку прогнозирования по сравнению со статическим методом. Исследование также показало, что сокращение пространства состояний посредством выбора главных компонентов может значительно сократить время принятия решения. Сравнительный анализ проводился с использованием таких моделей, как LSTM и Hidden Markov Model.

Полученные результаты работы могут быть использованы промышленными предприятиями и сервисными организациями, осуществляющими мониторинг критически важного оборудования и сталкивающимися с проблемой прогнозирования остаточного ресурса. Предложенный метод позволяет повысить эффективность принятия решений и сократить затраты на обслуживание оборудования, обеспечивая при этом точность прогнозов.

Abstract

Master's thesis, 41 pages, 4 figures, 50 sources, 61 formulas

Keywords: APPROXIMATE MARKOV CHAIN, SEQUENTIAL STATISTICAL DECISION RULE, PERFORMANCE CHARACTERISTICS CALCULATION, REMAINING USEFUL LIFE, DYNAMIC QUANTILE DISCRETIZATION.

The object of the research is approximate Markov chains and methods of sequential statistical decision making in problems of predicting the remaining life of equipment.

The subject of the research is the development and verification of a computational method for assessing performance characteristics based on approximate Markov chains for sequential statistical decision-making systems in problems of predicting the remaining life of industrial equipment.

The aim of this work is the development of a computationally efficient method for assessing performance characteristics based on approximated Markov chains for industrial equipment monitoring systems.

In the course of the work, the approximate Markov chain method and its application in sequential statistical decision making for the problem of predicting the remaining life of industrial equipment were developed. A computational model based on dynamic quantile discretization and PCA dimensionality reduction was proposed. The results show that adaptive updating of the sampling threshold every 100 cycles can reduce the prediction error compared with the static method. The study also showed that state space reduction through principal component selection can significantly reduce the decision making time. Comparative analysis was conducted using such models as LSTM and Hidden Markov Model.

The obtained results obtained can be used by industrial enterprises and service organizations monitoring critical equipment and facing the problem of predicting the remaining resource. The proposed method allows for increasing the efficiency of decision-making and reducing equipment maintenance costs, while ensuring the accuracy of forecasts.

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

RUL	Оставшийся полезный срок службы
LSTM	Сеть с длинной краткосрочной памятью
MAE	Средняя Абсолютная Ошибка
RMSE	Среднеквадратическая Ошибка
SVR	Метод Опорных Векторов для Регрессии
HMM	Скрытая Марковская Модель
FPR	Частота Ложных Срабатываний
PCA	Метод Главных Компонент
Recall@95%	Полнота при 95% точности
DTMC	Дискретно-временная цепь Маркова
CTMC	Непрерывная цепь Маркова
ICA	Метод независимых компонент
CVR	Коэффициент конверсии
KL	Дивергенция Кульбака-Лейблера

ВВЕДЕНИЕ

С наступлением эпохи «Индустрии 4.0» мониторинг состояния и прогнозирование остаточного срока службы ключевого оборудования, такого как авиационные двигатели и ветряные турбины, стали основными технологиями, обеспечивающими надежность оборудования и безопасность производства. По статистике Международного общества автоматизации, эффективное прогнозирование остаточного ресурса может сократить расходы на техническое обслуживание на 25–40%. Мы можем применить теорию последовательных статистических решений для динамического анализа временных рядов данных датчиков и построить вероятностную модель состояния деградации устройства.

Вычислительная сложность традиционных цепей Маркова растет экспоненциально в многомерных пространствах состояний и требует очень много времени. Другие существующие методы, такие как нейронные сети LSTM, вряд ли могут удовлетворить потребности в реагировании в реальном времени для обнаружения устройств. Статические методы дискретизации игнорируют дрейф распределения при ухудшении характеристик устройства и страдают от высокой потери производительности обобщения. Для решения этих задач мы разработали приближенный метод цепей Маркова. Инновация метода заключается в: разработке динамического квантильного алгоритма скользящего окна; создание модели связи на основе анализа главных компонент и цепей Маркова; и настройка оптимальной по Парето системы принятия решений. Экспериментальная проверка показала, что наш метод имеет очевидные преимущества по сравнению с существующими методами: он снижает вычислительную сложность и потери при обобщении, а также повышает вычислительную эффективность и точность.

Наши исследования могут предоставить услуги по предупреждению неисправностей авиационных двигателей в авиационной отрасли; его также можно применять для мониторинга редукторов ветряных турбин с целью продления срока службы оборудования; он также может помочь промышленным устройствам Интернета вещей достичь мгновенной оценки состояния работоспособности и т. д. Предполагается, что в ближайшем будущем его также можно будет применять в сценариях с высоким уровнем риска, таких как диагностика систем охлаждения атомных электростанций и мониторинг подшипников высокоскоростных поездов, обеспечивая теоретическую и техническую поддержку для создания управляемой и автономной интеллектуальной экосистемы обслуживания.

ГЛАВА 1

ВВЕДЕНИЕ И ОБЗОР ИССЛЕДОВАНИЯ

1.1 Предыстория и значимость исследования

В последние годы, в условиях взрывного роста объема информации и постоянного увеличения вычислительной мощности, моделирование и анализ сложных систем играют ключевую роль в современной науке, инженерных технологиях, социальной экономике и т. д.^[1]. В процессе принятия решений в сложных системах последовательное статистическое принятие решений играет важную роль в промышленном управлении, распределении ресурсов, оценке рисков и других областях благодаря своим динамическим оптимизационным характеристикам^[2]. В современных промышленных системах, чтобы обеспечить безопасную и надежную работу системы, обнаружение состояния здоровья и прогнозирование неисправности оборудования очень важны, особенно для ключевого оборудования, такого как авиационные двигатели, ветряные турбины и ядерные реакторы. Точное прогнозирование оставшегося срока службы может сформулировать стратегии обслуживания, снизить эксплуатационные расходы и избежать катастрофических сбоев^[3].

Последовательные статистические решения принимают решения на основе характеристик времени наблюдаемых данных. Например, последовательные статистические решения могут использовать информацию о времени датчика данных в управлении здоровьем оборудования для улучшения своевременности и точности прогнозирования разломов^[4]. Тем не менее, в действительности, промышленные системы часто имеют высокомерные государственные пространства и в то же время имеют сложные отношения переноса состояний и динамические условия труда, которые трудно конкурировать с традиционными последовательными статистическими методами принятия решений^[5].

Цепь Маркова — это случайный процесс. Широко используется в динамическом моделировании системы^[6]. Благодаря тому, что «будущее состояние» зависит только от текущего состояния, оно имеет преимущества при анализе производительности: его матрица вероятностей перехода состояний удобна для математических вычислений и может использоваться для расчета показателей производительности, таких как вероятность устойчивого состояния системы и среднее время первого прохождения. Он может хорошо описывать закон перехода состояний динамических систем. Однако точные цепи Маркова часто испытывают трудности при решении задач с многомерными пространствами состояний и сложными переходными структурами. Поэтому мы можем использовать некоторые приближенные методы (такие как агрегация пространства состояний, уменьшение размерности параметров и т. д.) для

упрощения цепи Маркова и уменьшаем вычислительную сложность, обеспечивая точность прогнозирования и применяем метод последовательного статистического принятия последовательных статистических решений Маркова в промышленных системах в промышленных системах.

1.2 Текущее состояние исследований в стране и за рубежом

Последовательная статистическая теория принятия решений, возникшая из метода последовательного анализа 1950-х годов. Ранние исследования были сосредоточены на оптимальной проблеме остановки теста гипотезы и оценки параметров. На основе теста коэффициента последовательных вероятностей Уолда (SPRT)^[7]. Позже ученые изучали последовательные методы принятия решений, такие как проблема с мультирузом игровой автоматы, оптимальная теория времени остановки и динамическое планирование^[8]. С появлением и популярностью обучения подкрепления и байесовских методов ученые объединили динамическое программирование, процессы Маркова и технологии, управляемые данными, для формирования междисциплинарного гибридного направления исследования.

Метод аппроксимации цепи Маркова в основном включает в себя четыре метода: дискретизация пространства состояния, понижение модели, агрегация и параметризованные приближения. Метод дискретизации пространства состояния уменьшает вычислительную сложность с помощью дискретного пространства непрерывного состояния в конечное дискретное состояние^[9]. В основном он включает в себя равные расстояния, равную частоту, дискретизацию кластера и т.д. Методы агрегации включают агрегацию по состоянию и времени. Агрегация состояний применяется к моделированию сложных систем на основе сходства признаков, например, кластеризации методом К-средних. Временное агрегирование может применяться для расчета вероятности банкротства в области страховой математики, приближенного метода кусочно-детерминированного марковского процесса и преобразования временной модели из непрерывной в дискретную с помощью итеративной аппроксимации и технологии сглаживания.^[10] Технология сокращения порядка моделей снижает размеры переменных состояния с помощью методов уменьшения размерности, таких как анализ основных компонентов и анализ независимых компонентов^[11]. Приблизительное распределение вероятностей переноса с помощью параметризованных моделей. Хотя эти методы широко используются, все еще существуют такие проблемы, как трудности в контроле ошибок приближения и низкая вычислительная эффективность.

Чтобы оценить и оптимизировать систему решений, нам необходимо выполнить расчеты характеристик производительности. Традиционные методы

расчета включают в себя: непосредственное поиск индикаторов производительности с помощью математических методов^[12], с использованием численных методов, таких как численная интеграция и операции матрицы для расчета индикаторов производительности^[13], и использование случайных методов моделирования, таких как моделирование Монте -Карло для оценки показателей производительности^[14]. В последние годы также разработаны методы расчета характеристик эффективности, управляемые данными. Существующие методы сталкиваются с проблемами эффективности вычислений и баланса точности в крупномасштабных, высокоразмерных динамических системах.

1.3 Содержание исследования и инновация

Сначала мы используем динамическую квантильную дискретизацию, анализ основных компонентов и методы уменьшения размерности для создания приблизительной модели цепи Маркова, которая подходит для динамических высокоразмерных систем. Затем последовательная статистическая модель принятия решений для управления оборудованием создается на основе приблизительной модели цепи Маркова. Затем мы разрабатываем эффективные алгоритмы расчета функций эффективности, включая точность решения, задержку принятия решений, сложность расчета и т.д. В экспериментальной части мы выбрали набор данных C-MAPSS NASA для экспериментальной проверки и анализа для анализа эффективности и практичности нашего метода.

Наше инновация заключается в том, что он предлагает адаптивный механизм квантильного обновления, который может эффективно адаптироваться к старению оборудования. Чтобы построить приблизительную структуру цепи Маркова, мы объединяем технологии дискретизации пространства и размерности состояния. Предлагается метод оптимизации стратегии принятия решений, основанный на передовом Парето-фронт, который достигает оптимального баланса точности и эффективности. Заменив метод проверки набора данных, мы докажем способность обобщения и практичность нашего метода.

ГЛАВА 2

СВЯЗАННЫЕ ОСНОВНЫЕ ТЕОРИИ

2.1 Базовая теория цепей Маркова

Существует пространство состояний $S=\{1,...,N\}$, охватывает все возможные состояния цепи Маркова, случайные величины $X_n \in S$ и дискретный во времени случайный процесс, удовлетворяющий свойству Маркова, а именно дискретная во времени цепь Маркова (DTMC). Марковская характеристика заключается в том, что текущее состояние времени связано только с предыдущим состоянием времени и не имеет ничего общего с другими состояниями времени^[15], то есть:

$$P\{X_{n+1} = j | X_n = i\} = P\{X_1 = j | X_0 = i\} = p_{ij} \quad (2-1-1)$$

Для цепи Маркова $\{X_n, n \geq 0\}$ вероятность одношагового перехода из состояния i в состояние j равна:

$$p_{ij}(n) = P\{X_{n+1} = j | X_n = i\} \quad (2-1-2)$$

Если вероятность перехода не зависит от времени n , то есть $p_{ij}(n) = p_{ij}$ выполняется для всех n , то цепь Маркова называется однородной цепью Маркова.

Для однородной цепи Маркова с конечным пространством состояний $S=\{1,...,N\}$ ее матрица вероятностей перехода P имеет вид:

$$P = \begin{pmatrix} p_{11} & \cdots & p_{1N} \\ \vdots & \ddots & \vdots \\ p_{N1} & \cdots & p_{NN} \end{pmatrix} \quad (2-1-3)$$

Где $p_{ij} \geq 0$, и для $\forall i$, удовлетворяет:

$$\sum_{j=1}^N p_{ij} = 1 \quad (2-1-4)$$

Имеем формулу Чепмена-Колмогорова:

$$p_{ij}^{(n+m)} = \sum_{k=1}^N p_{ik}^{(n)} p_{kj}^{(m)} \quad (2-1-5)$$

$$p^{(n+m)} = p^n p^m = p^m p^n \quad (2-1-6)$$

Определим случайную величину $X(t)$, случайный процесс, который удовлетворяет свойству Маркова в непрерывном времени, называется непрерывной временной цепью Маркова (CTMC).

Аналогично определяется матрица вероятностей перехода, где элемент в i -й строке и j -м столбце матрицы представляет собой вероятность перехода из i в j через время t :

$$P_t = (P_{ij}(t))_{ij} \quad (2-1-7)$$

Существует также формула Чепмена-Колмогорова:

$$P(s + t) = P(s)P(t) = P(t)P(s) \quad (2-1-8)$$

$$P_{ij}(s + t) = \sum_{k \in S} P_{ik}(s)P_{kj}(t) \quad (2-1-9)$$

Определим матрицу скорости передачи Q_{ij} , Нondiagonalные элементы $q_{ij} \geq 0 (i \neq j)$, Диагональные элементы $q_{ii} \leq 0$, И сумма каждой строки матрицы равна 0:

$$\sum_j q_{ij} = 0 \quad (2-1-10)$$

Его матрица вероятностей перехода удовлетворяет условиям:

$$P_{ij}(t) = e^{Q_{ij}t} = \sum_{k=0}^{\infty} \frac{Q_{ij}^k t^k}{k!} \quad (2-1-11)$$

Удовлетворяет уравнению Колмогорова: Впередное уравнение:

$$\frac{d}{dt} p_{ij}(t) = \sum_{k \in S} p_{ik}(t) q_{kj} \quad (2-1-12)$$

Обратное уравнение:

$$\frac{d}{dt} p_{ij}(t) = \sum_{k \in S} q_{ik} p_{kj}(t) \quad (2-1-13)$$

Цепь Маркова эргодична. Для любых двух состояний i и j в пространстве состояний S существует $p_{ij}(n) > 0$, то есть из любого состояния i можно достичь состояния j за n шагов. Он неприводим, и все состояния связаны и достижимы друг для друга^[16]. Он обладает характер частого возврата. Когда время стремится к бесконечности, вероятность достижения другого состояния в первый раз, начиная с любого состояния, не равна 0. Период $d(i)$ относится к наибольшему общему делителю времени возврата в состояние. Когда $d(i) > 1$, состояние i называется периодическим, а когда $d(i)=1$, состояние i называется непериодическим^[17]. Если цепь Маркова неприводима, непериодична и у неё есть характер частого возврата, то по мере того, как время стремится к бесконечности, распределение состояний цепи Маркова близко к стационарному распределению

$$\lim_{n \rightarrow \infty} P(X_n = j) = \pi_j \quad (2-1-14)$$

, и существует единственное стационарное распределение^[18]

$$\pi = (\pi_j), j \in S \quad \pi_j = \sum_{i \in S} \pi_i p_{ij} \quad (2-1-15)$$

, и

$$\sum_{j \in S} \pi_j = 1, \pi_j \geq 0 \quad (2-1-16)$$

2.2 Последовательная статистическая теория принятия

решений

2.2.1 Статистика заказов при условиях Маркова

Предположим, что $X_1, X_2, \dots, X_n$ являются выборками популяции X , и отсортируем наблюдаемые значения выборок от меньших к большим, $x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)}$, $X_{(1)}$ является минимальным значением, $X_{(n)}$ является максимальным значением, а $X_{(i)}$ является статистикой i -го порядка^[19]. Выражение плотности распределения вероятностей статистики k -го порядка $X_{(k)}$ имеет вид:

$$f_{x_{(k)}}(x) = \frac{n!}{(k-1)!(n-k)!} F(x)^{k-1} [1 - F(x)]^{n-k} f(x) \quad (2-2-1)$$

Порядковая статистика в условиях Маркова: изменения состояния могут влиять на порядок значений выборки^[20]. Например, в модели случайного роста, если процесс роста особей удовлетворяет свойству Маркова, то порядок размера (статистика порядка) особей может меняться со временем. Новые добавленные значения выборки при каждом переходе состояния также изменяют статистику заказа. Статистика заказов в условиях Маркова имеет условную независимость, и обновление статистики заказов в следующий момент времени зависит только от текущей выборочной ситуации^[21].

$$\Pr(X_k | X_{k-1}, \dots, X_1) = \Pr(X_k | X_{k-1}) \quad (2-2-2)$$

Сочетание этих двух методов может применяться для анализа надежности оборудования (минимальная статистика порядка компонентов оборудования определяет время отказа всей системы^[22]). Его также можно использовать для анализа экстремальных порядковых статистик доходности активов с целью анализа максимально возможных убытков или финансового риска.

2.2.2 Последовательная статистическая модель принятия

решений

Последовательное статистическое принятие решений предполагает принятие решений на основе совокупной информации из последовательности наблюдений. Структура последовательной статистической модели принятия решений выглядит следующим образом: сначала наблюдение выполняется в каждый момент времени t для получения значения наблюдения X_t . Затем принимаются решения δ_t на основе истории наблюдений $\mathcal{H}_t = \{X_1, X_2, \dots, X_t\}$. Затем установим оптимальное время остановки наблюдения τ и, наконец, разработаем целевую функцию для минимизации ожидаемых потерь или максимизации ожидаемого выигрыша^[23].

Предположим, что случайный процесс $\{X_t, t \geq 1\}$, $\mathcal{T}$ представляет все возможные времена остановки, а задача оптимального времени остановки заключается в решении следующей формулы:

$$\tau^* = \arg \max_{\tau \in \mathcal{T}} E[R(X_\tau, \tau)] \quad (2-2-3)$$

$R(x, t)$ представляет собой функцию прибыли от прекращения наблюдения в момент времени t и получения значения наблюдения x .

Последовательные статистические решения и приближительные цепочки Маркова могут быть объединены в динамических неопределенных средах и могут использоваться для расчета характеристик крайней стоимости и эффективности принятия решений во время переноса состояния. Эта модель уменьшает сложность расчета с помощью методов приближения, таких как дискретизация пространства состояния и снижение размеров матрицы в пространстве и переноса^[24], и использует последовательную статистику для захвата характеристик хвоста и квантильной эффективности последовательности результатов принятия решения^[25].

2.3 Метод расчета характеристик эффективности

2.3.1 Общие показатели эффективности

В этом разделе мы перечислим некоторые ключевые показатели эффективности в последовательной статистической структуре принятия решений на основе приближенных цепей Маркова, а также их теоретическую основу и сценарии применения. Первый показатель — это показатель точности решения, который в основном включает в себя показатель ошибок, показатель ложных тревог, показатель точности и показатель полноты. задержка принятия решения и вычислительная сложность^[26]. Некоторые определения и формулы расчета приведены в следующей таблице:

Таблица 2.3.1 - Показатели оценки, используемые для сравнения

Индикаторы оценки	Определение	Формула расчета
Скорость отзыва ошибок TPR (Recall@95%)	Доля правильно предсказанных предвестников отказов	$\frac{TP}{(TP + FN)}$
Коэффициент ложных срабатываний(FPR)	Доля нормального статуса, ошибочно принимаемого за ошибку	$\frac{FP}{(FP + TN)}$

Точность	Доля нормальных состояний, предсказанных как правильные	$\frac{TP}{(TP + FP)}$
Задержка принятия решения	Среднее время принятия одного решения	Рассчитать время с помощью кода
RUL предсказывает MAE	Средняя абсолютная ошибка прогнозирования остаточного срока службы	$MAE = \frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i $

TP представляет собой количество образцов, которые на самом деле являются ошибочными и правильно предсказаны как ошибочные, а FN представляет собой количество образцов, которые на самом деле являются ошибочными и неправильно предсказаны как нормальные. FP представляет собой количество образцов, которые на самом деле нормальны, но ошибочно оценены как ненормальные, а TN представляет собой количество образцов, которые на самом деле нормальны, но правильно оценены как нормальные^[27].

2.3.2 Традиционный метод расчета характеристик эффективности

На основе описанных выше характеристик производительности перечислим три классических метода расчета характеристик производительности и их применение в приближенных цепях Маркова.

Численные методы — это методы, которые рассчитывают показатели эффективности с помощью численного интегрирования, матричных операций и других методов. Они подходят для сложных систем и позволяют получать приблизительные значения показателей эффективности путем численной аппроксимации. Например, метод трапецеидального интегрирования делит интервал интегрирования сложной функции f на несколько небольших интервалов и применяет метод трапецеидального интегрирования к каждому небольшому интервалу для получения результата^[28]. Основная формула:

$$\int_a^b f(x)dx \approx \frac{b-a}{2} [f(a) + f(b)] \quad (2-3-1)$$

Мы часто используем обращение матрицы для решения линейных уравнений. Пусть A — матрица коэффициентов, x — неизвестный вектор, а b — известный вектор. Для системы линейных уравнений $Ax = b$ мы можем выполнить обращение матрицы и получить^[29]:

$$x = A^{-1}b \quad (2-3-2)$$

Классический метод моделирования Монте-Карло оценивает показатели

эффективности путем случайной выборки. Основная идея заключается в использовании случайной выборки для аппроксимации ожидаемого значения или распределения вероятностей сложной системы. Пусть $g(\cdot)$ — функция производительности, используемая для оценки производительности каждого образца X_i , а M — количество симуляций. Чтобы обеспечить точность оценки, количество симуляций часто должно быть достаточно большим. Основная формула^[30]:

$$\hat{\mu} = \frac{1}{M} \sum_{i=1}^M g(X_i) \quad (2-3-3)$$

Точность моделирования Монте-Карло можно оценить, рассчитав предполагаемую дисперсию:

$$\hat{\sigma}^2 = \frac{1}{M-1} \sum_{i=1}^M (g(X_i) - \hat{\mu})^2 \quad (2-3-4)$$

Если мы хотим оценить время отклика сложной системы, мы можем сначала определить функцию производительности, установить $g(X_i)$ в качестве времени отклика i -го моделирования, затем сгенерировать M случайных выборок $X_1, X_2, \dots, X_M$, а затем вычислить среднюю оценку времени отклика $\hat{\mu}$ по формуле и, наконец, вычислить предполагаемую дисперсию $\hat{\sigma}^2$ для оценки надежности оценки.

Последний метод называется аналитическим решением, при котором показатели эффективности рассчитываются напрямую посредством математического вывода. Применимо к простым линейным системам. Предположим, что имеется простая линейная система первого порядка:

$$y(t) = ax(t) + b \quad (2-3-5)$$

$y(t)$ представляет выход системы, $x(t)$ представляет вход системы, а a и b являются параметрами системы. На этом этапе мы хотим рассчитать установившуюся ошибку этой системы. Для простой линейной системы формула установившейся ошибки имеет вид^[31]:

$$e_{ss} = \lim_{t \rightarrow \infty} (r(t) - y(t)) \quad (2-3-6)$$

$r(t)$ — опорный вход. Для нашей линейной системы первого порядка установившаяся ошибка равна:

$$e_{ss} = \frac{1}{1+a} \quad (2-3-7)$$

Таблица 2.3.2 - Сравнение трех традиционных методов

Метод	Преимущества	Ограничение	Совместимость с приближенными цепями Маркова

Динамическое программирование	Глобальное оптимальное решение может быть получено	Высокоразмерные состояния невозможны	Объединение состояний агрегации и аппроксимация функции
Моделирование Монте-Карло	Не требуется никаких модельных предположений, подходит для сложных систем	Требуется большое количество образцов, большая дисперсия	Выборка по важности фокусируется на траекториях высокой вероятности
Аналитическое решение	Точные результаты и высокая вычислительная эффективность	Применимо только к специальным конструкциям	Построение приближенного о процесса рождения-смерти с использованием грубой детализации состояния

2.4 Обзор методов аппроксимации

2.4.1 Распространенные методы аппроксимации

Технология сжатия пространства состояний относится к процессу сокращения числа состояний N посредством кластеризации или агрегации таким образом, чтобы исходный набор состояний $S = \{s_1, \dots, s_N\}$ отображался в сжатом наборе состояний $S' = \{s'_1, \dots, s'_K\}$ ($K \ll N$). К ним относятся методы кластеризации, методы квантования и методы снижения размерности. Метод кластеризации заключается в объединении схожих состояний с использованием таких методов, как K-средние и иерархическая кластеризация^[32]. Метод квантования заключается в дискретизации непрерывного пространства состояний на конечные уровни квантования. Метод снижения размерности заключается в использовании таких методов, как PCA и ICA, для снижения размерности переменных состояния. Сжатие пространства состояний может значительно снизить вычислительную сложность, но может привести к отклонениям в слиянии, и для проверки точности сжатия необходим анализ ошибок.

Технология аппроксимации параметров заключается в снижении вычислительной сложности за счет упрощения зависимостей параметров в

модели или введения приближенных распределений параметров. К ним относятся такие методы, как оценка плотности ядра, модели гауссовской смеси и байесовские сети. Пусть K — функция ядра, h — параметр полосы пропускания, а x_i — точка данных. Формула для оценки плотности вероятности с использованием функции ядра:

$$\widehat{f(x)} = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (2-3-8)$$

Модель гауссовой смеси представляет собой линейную комбинацию нескольких гауссовых распределений для аппроксимации сложных распределений вероятностей^[33]. Байесовские сети используют направленные ациклические графы для представления условных зависимостей между переменными. Расчет приближения параметров эффективен и может использоваться для онлайн-оценки, но сложность сопоставления моментов высокого порядка также очень высока, а ошибка расширения Тейлора относительно большая в сильных нелинейных сценариях.

Таблица 2.4.1 - Сценарии применения трех методов аппроксимации

Приблизительный метод	Приложение	Практическое применение
Параметр аппроксимации	Модели с низкой чувствительностью параметров и неопределенными параметрами	Оценка финансового риска. Сети химических реакций в биологических системах
Сжатие пространства состояний	Крупномасштабное дискретное пространство состояний, проблемы принятия решений в реальном времени	Крупномасштабные сети очередей (например, системы связи)

ГЛАВА 3

ПОСТРОЕНИЕ МОДЕЛИ, АППРОКСИМИРУЮЩЕЙ ЦЕПЬ МАРКОВА

3.1 Введение в проблему и основные предположения

В соответствии с исследовательским фоном и требованиями статьи в качестве объекта исследования мы используем набор данных моделирования деградации газотурбинного двигателя NASA. Этот набор данных моделирует процесс работы двигателя от нормального состояния до ухудшения характеристик и содержит данные обнаружения с нескольких датчиков с корреляцией временных рядов (температура, давление, скорость и т. д.)^[34]. Из этих данных мы можем извлечь экстремальные значения или квантили, чтобы оценить состояние оборудования и решить, необходимы ли операции по техническому обслуживанию. (Если данные датчика превышают исторический квантильный порог, инициируется операция по техническому обслуживанию.)

Мы изучаем динамические системы с многосенсорным мониторингом. Пусть d — общее количество датчиков. Для каждого момента времени t система получает вектор данных датчика:

$$x_t = [x_t^1, x_t^2, \dots, x_t^d]^T \quad (3-1-1)$$

Наша задача — построить приближенную марковскую модель, показывающую, как состояние системы меняется с течением времени, и спрогнозировать на основе этой модели остаточный срок службы (RUL) системы.

Для упрощения сложного процесса деградации мы делаем несколько предположений: во-первых, мы предполагаем, что переход состояния деградации зависит только от текущего состояния и состояния предыдущих $k-1$ моментов, то есть:

$$P(s_{t+1} | s_t, s_{t-1}, \dots, s_0) \approx P(s_{t+1} | s_t, s_{t-1}) \quad (k = 2) \quad (3-1-2)$$

Во-вторых, мы предполагаем, что матрица вероятностей перехода состояний остается постоянной в течение одного полета. Это предположение требует, чтобы параметры двигателя были относительно стабильными в течение одного полета и не было никаких внезапных внешних помех^[34]. Если задача подразумевает переключение между рабочими состояниями, требуется сегментированное моделирование. Наконец, мы предполагаем, что непрерывные данные датчиков могут быть отображены в конечные дискретные состояния посредством квантильного разбиения. Распределение значений датчиков

данных C-MAPSS выглядит следующим образом:

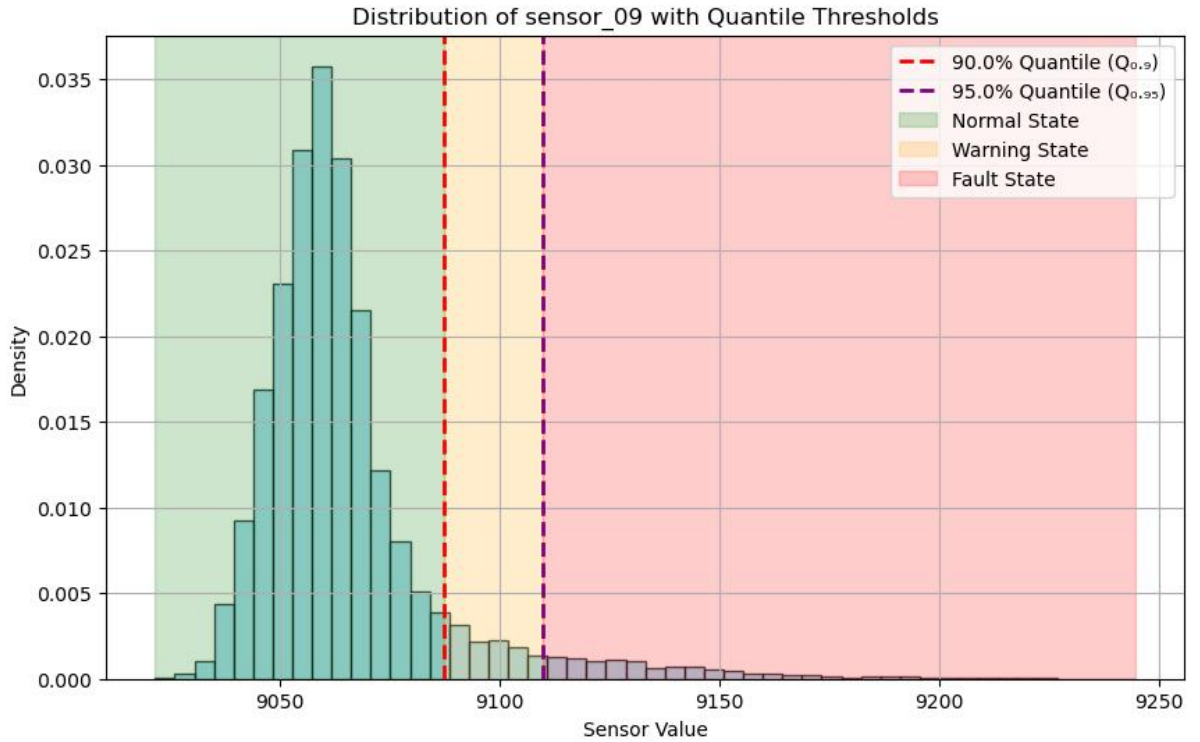


Рисунок 3.1.1 - Распределение значений датчика с квантильными порогами

Видно, что распределение значений датчиков демонстрирует значительную хвостовую особенность, а квантильное деление выделяет нормальные и аномальные диапазоны, а именно:

$$S_i = \begin{cases} \text{нормальный, } x \leq Q_{0.9}(S_i) \\ \text{Раннее предупреждение, } Q_{0.9}(S_i) < x \leq Q_{0.95}(S_i) \\ \text{Не работает, } x > Q_{0.95}(S_i) \end{cases} \quad (3-1-3)$$

Мы пересчитываем порог квантиля каждые 100 раз и настраиваем механизм динамического обновления для адаптации к старению устройства.^[35]

3.2 Автокорреляционный анализ и марковская проверка

Для временных данных $\{x_t\}$, сгенерированных датчиком, функция автокорреляции после задержки k равна:

$$\rho(k) = \frac{E[(x_t - \mu)(x_{t-k} - \mu)]}{\sigma^2} \quad (3-2-1)$$

μ — ожидаемое значение последовательности, а σ^2 — дисперсия последовательности.

В практических приложениях для расчета часто используется выборочная автокорреляционная функция ($\bar{x}$ представляет выборочное среднее значение последовательности):

$$\widehat{\rho(k)} = \frac{\sum_{t=k+1}^n (x_t - \bar{x})(x_{t-k} - \bar{x})}{\sum_{t=1}^n (x_t - \bar{x})^2} \quad (3-2-2)$$

Автокорреляционный анализ подмножества FD001 данных C-MAPSS, дает следующий рисунок:

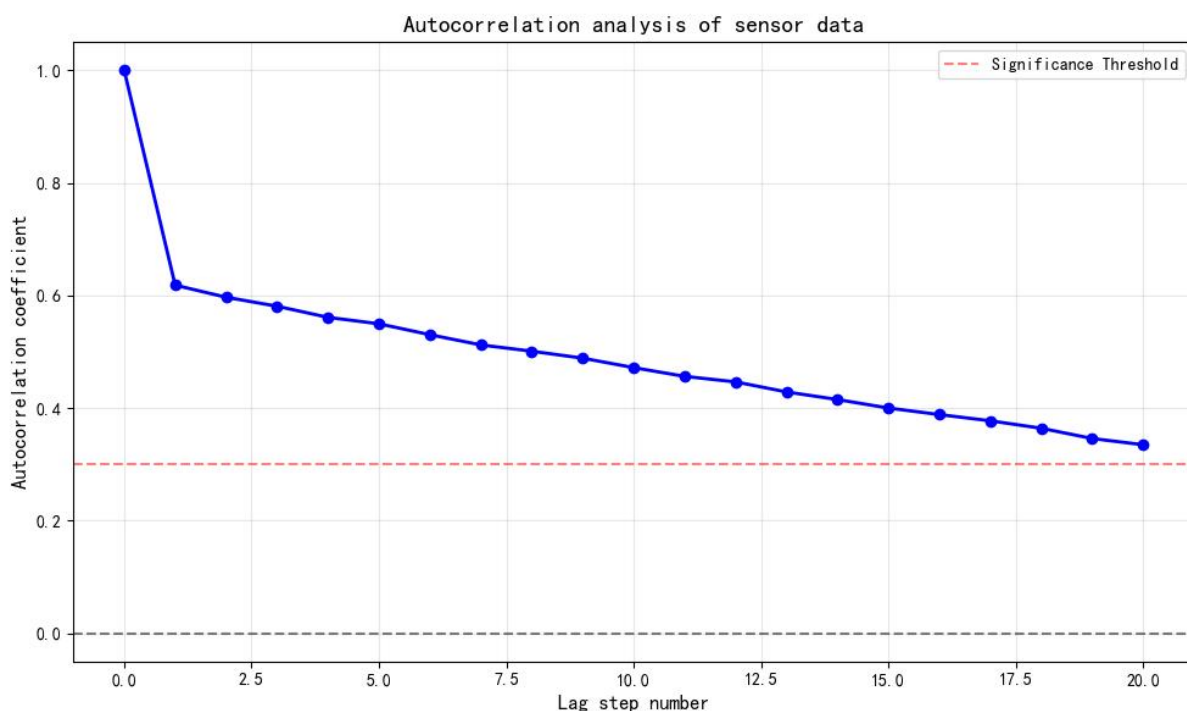


Рисунок 3.2.1 - Изменение коэффициента автокорреляции данных датчика в зависимости от количества шагов запаздывания

Видно, что коэффициент автокорреляции датчика значительно уменьшается после задержки в 1,2 шага, что согласуется с основным предположением цепи Маркова о том, что текущее состояние зависит только от предыдущего и двух предыдущих состояний и не имеет ничего общего с более ранними состояниями. Данные имеют краткосрочную актуальность и не имеют долговременной памяти.

3.3 Метод динамической квантильной дискретизации

Традиционная квантильная дискретизация заключается в разделении непрерывной переменной x на K интервалов в соответствии с ранее заданными квантилями $q_1, q_2, \dots, q_{K-1}$:

$$S_k = \begin{cases} (-\infty, q_1], & k = 1 \\ (q_{k-1}, q_k], & k = 2, \dots, K-1 \\ (q_{K-1}, +\infty), & k = K \end{cases} \quad (3-3-1)$$

Соответствующая функция кодирования состояния определяется как:

$$s(x) = k, \text{ if } x \in S_k \quad (3-3-2)$$

Чтобы адаптироваться к изменениям в распределении данных по мере

старения устройства, мы предлагаем динамический механизм обновления квантилей. Чтобы адаптироваться к изменениям в распределении данных в процессе старения устройства, мы предлагаем механизм динамического обновления квантиля: мы вводим поток данных датчика $\{x_t\}$, частоту обновления Δt и количество состояний k , используем первые Δt выборки для расчета начального квантиля $q_k(0)$ и используем выборки в скользящем окне $[t - \Delta t + 1, t]$, вычисляем новый квантиль:

$$q_k(t) = \text{Квантиль}(x_{\{t-\Delta t+1:t\}}, k/K) \quad (3-3-3)$$

и обновляем кодировку состояния:

$$s_t = \arg \min_k |x_t - q_k(t)| \quad (3-3-4)$$

выводим порог динамического квантиля.

Наш динамический механизм обновления квантилей позволяет отслеживать изменения в распределении данных в режиме реального времени, то есть модель всегда соответствует текущим характеристикам данных. Благодаря обновлению скользящего окна влияние выбросов на оценку квантиля снижается. Он подходит для сценариев приложений в реальном времени и может быстро реагировать на изменения в распределении данных в режиме онлайн.

3.4 Снижение размерности главного компонентного анализа

Анализ главных компонент подразумевает преобразование исходного многомерного пространства признаков в маломерное пространство посредством линейного преобразования, сохраняя наиболее важную информацию в данных. Для данной матрицы данных $X \in \mathbb{R}^{n \times d}$ целью анализа главных компонент является нахождение ортогональной матрицы преобразования $W \in \mathbb{R}^{d \times k}$ ($d < k$) таким образом, чтобы матрица данных с уменьшенной размерностью $Y \in \mathbb{R}^{n \times k}$ охватывала основные характеристики данных^[36]:

$$Y = XW \quad (3-4-1)$$

Процесс решения анализа главных компонент включает четыре этапа: центрирование, вычисление ковариационной матрицы, разложение собственных значений и выбор главных компонент. Центрирование заключается в центрировании матрицы данных X для получения $\tilde{X} = X - 1\mu^T$, где μ — вектор среднего значения столбца^[37]. Затем вычисляем ковариационную матрицу центрированной матрицы данных:

$$C = \frac{1}{n-1} \tilde{X}^T \tilde{X} \quad (3-4-2)$$

Далее ковариационная матрица C разлагается для получения собственного значения λ_i и соответствующего собственного вектора v_i , где собственные

значения располагаются в порядке убывания:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d \quad (3-4-3)$$

Наконец, первые k главных компонент используются для формирования матрицы преобразования:

$$W = [v_1, v_2, \dots, v_k] \quad (3-4-4)$$

При выборе числа главных компонент мы в основном имеем в виду кумулятивную ставку вклада дисперсии:

$$\text{CVR}(k) = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^d \lambda_i} \quad (3-4-5)$$

Выбираем наименьшее k такое, что: $\text{CVR}(k) \geq 0.85$.

Тепловая карта матрицы переноса цепи Маркова, полученная после динамической квантильной дискретизации и снижения размерности PCA на FD001, выглядит следующим образом:

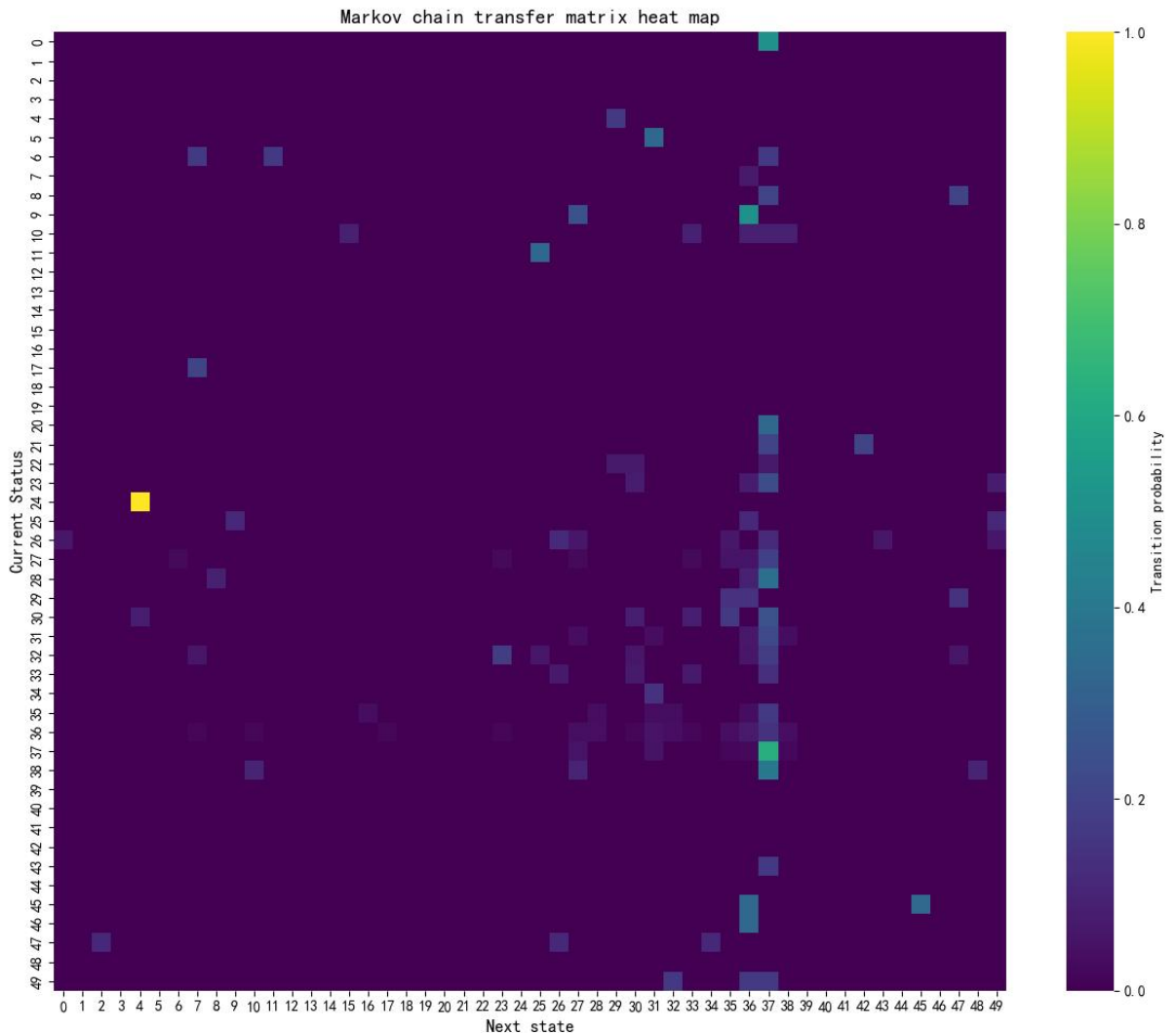


Рисунок 3.1.1 - Тепловая карта матрицы переноса цепи Маркова

Как видно из рисунка, более 95% площади имеет темно-фиолетовый цвет, а вероятность большинства переходов состояний очень низкая. Последовательный перенос доминирует в промежуточном процессе, существуют единичные скачки из состояния 0 в 22, а состояние 49 является поглощающим состоянием.

3.5 Оценка вероятностей перехода

Мы используем метод оценки плотности ядра для оценки вероятностей перехода состояний. Расчетная формула для перехода из состояния i в состояние j :

$$\hat{p}_{ij} = \frac{\sum_{t=1}^{n-1} K_h(s_t - i) K_h(s_{t+1} - j)}{\sum_{t=1}^{n-1} K_h(s_t - i)} \quad (3-5-1)$$

Среди них $K_h(\cdot)$ — это функция ядра с полосой пропускания h ^[38], а обычно используемая формула функции ядра Гаусса имеет вид:

$$K_h(u) = \frac{1}{\sqrt{2\pi}h} \exp\left(-\frac{u^2}{2h^2}\right) \quad (3-5-2)$$

Чтобы повысить вычислительную эффективность и уменьшить шумовые помехи, мы выполняем усечение с низкой вероятностью для оценочной вероятности перехода: мы вводим оценочную матрицу вероятности перехода $\hat{P}$, порог усечения ε и выводим усеченную матрицу вероятности перехода P . Конкретные шаги алгоритма следующие: первый шаг — вычисление суммы строк до усечения: $S_{before} = \sum_j \hat{P}_{ij}$; второй шаг — отсечение элементов с низкой вероятностью: $P_{ij} = \hat{P}_{ij}$ если $\hat{P}_{ij} \geq \varepsilon$ иначе 0; третий шаг — вычисление суммы строки после усечения: $S_{after} = \sum_j P_{ij}$; четвертый шаг — расчет коэффициента компенсации: $\alpha = S_{before}/S_{after}$; окончательно перенормировать: $P_{ij} = \alpha \times P_{ij}$.

После завершения оценки мы проверяем, удовлетворяет ли построенная матрица вероятностей перехода следующим свойствам: $p_{ij} \geq 0$ справедливо для всех i, j . $\sum_j p_{ij} = 1$ справедливо для всех i . Между любыми двумя состояниями существует достижимый путь. Наибольший общий делитель циклов перехода состояний равен 1.

3.6 Проверка и анализ модели

Мы используем дивергенцию KL для измерения соответствия между приближенной моделью и наблюдаемыми данными, пусть P представляет наблюдаемое распределение, а Q представляет прогнозируемое распределение модели. Выражение следующее:

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \quad (3-6-1)$$

Мы применяем метод перекрестной проверки временных рядов и делим первые 70% данных временного ряда на обучающий набор. Средние 15% данных используются в качестве проверочного набора. Последние 15% данных используются в качестве тестового набора.

Далее анализируется чувствительность модели к изменению параметров и проверяется число состояний $K \in \{3,4,5,6,7\}$. Проверяются частоты обновления $\Delta t \in \{50,100,150,200\}$. Была протестирована размерность РСА $k \in \{3,4,5,6\}$.

ГЛАВА 4

ПОСЛЕДОВАТЕЛЬНАЯ СТАТИСТИЧЕСКАЯ МОДЕЛЬ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ АППРОКСИМИРУЮЩЕЙ ЦЕПИ МАРКОВА

4.1 Построение модели последовательного статистического решения

Согласно приближенной цепи Маркова, построенной в предыдущей главе, мы определяем пространство состояний системы как: $S = \{s_1, s_2, \dots, s_N\}$. Каждое состояние s_i соответствует дискретному вектору состояний после уменьшения размерности, а размер пространства состояний равен $N = K^k$, где K — число дискретных состояний в каждом измерении, а k — измерение после уменьшения размерности PCA.

Пространство решений для прогнозирования оставшегося срока службы включает: $A = \{\text{продолжение эксплуатации, плановое техническое обслуживание, аварийное отключение}\}$. Интервал прогнозирования RUL, соответствующий каждому пространству принятия решений:

$$\begin{cases} \text{Продолжение работы: } RUL > T_1 \\ \text{Плановое техническое обслуживание: } T_2 < RUL \leq T_2 \\ \text{Аварийное отключение: } RUL \leq T_2 \end{cases} \quad (4-1-1)$$

T_1 и T_2 — это предустановленные пороговые значения.

Развернуть функцию вознаграждения за состояние-действие $R(s, a)$:

$$R(s, a) \begin{cases} R_{\text{normal}} - C_{\text{operation}} & \text{Если } a \text{ продолжено и правильно} \\ - C_{\text{maintenance}} & \text{Если это плановое техническое обслуживание и оно правильное} \\ - C_{\text{emergency}} & \text{Если } a \text{ — это аварийная остановка и она правильная} \\ - C_{\text{false}} & \text{Если решение неверное} \end{cases} \quad (4-1-2)$$

Мы применяем метод динамического программирования для разработки оптимальной стратегии принятия решений. Функция ценности^[39]:

$$V(s) = \max_a [R(s, a) + \gamma \sum_s P(s|s, a)V(s)] \quad (4-1-3)$$

где $\gamma \in [0, 1]$ — коэффициент дисконтирования.

4.2 Алгоритм прогнозирования оставшегося срока службы

Мы устанавливаем текущее состояние как s_t и вычисляем RUL на основе времени поглощения цепи Маркова. Мы предполагаем, что состояние отказа — s_f , а ожидаемое время поглощения, начиная с состояния s_i , удовлетворяет условию^[40]:

$$E[T_i] = 1 + \sum_{j \neq f} p_{ij} E[T_j] \quad (4-2-1)$$

Это образует линейную систему уравнений:

$$(I - Q)t = 1 \quad (4-2-2)$$

Q-матрица переноса после удаления поглощающего состояния,

$$t = [E[T_1], E[T_2], \dots, E[T_{N-1}]]^T \quad (4-2-3)$$

Затем рассчитывается распределение вероятностей RUL. Пусть $P_{ij}^{(n)}$ — вероятность перехода на n-м шаге, $T_i(n)$ представляет собой вероятность достижения поглощающего состояния в первый раз на шаге n, начиная с состояния i, что выражается как:

$$T_i(n) = \sum_j P_{ij}^{(n-1)} p_{jf} \quad (4-2-4)$$

В практических приложениях часто возникает необходимость в динамическом обновлении RUL на основе новых данных наблюдений. Динамический алгоритм прогнозирования RUL разработан следующим образом: мы вводим последовательность наблюдений $\{x_1, x_2, \dots, x_t\}$ и матрицу переноса P. Выводим прогнозы RUL и доверительные интервалы. Сначала выполните кодирование состояния: $s_t = \text{StateEncoding}(x_t)$; затем вычислить распределение вероятностей текущего состояния: $\pi_t = \text{UpdateBelief}(\pi_{t-1}, s_t, P)$; затем вычислить ожидаемый RUL: $RUL_t = \sum_i \pi_{t(i)} \times E[T_i]$; затем вычислить распределение RUL: $F_{RUL(n)} = \sum_i \pi_{t(i)} \times T_{i(n)}$; наконец, вычислите доверительный интервал:

$$[RUL_lower, RUL_upper] = \text{ConfidenceInterval}(F_{RUL}, \alpha) \quad (4-2-5)$$

4.3 Метод расчета эксплуатационных характеристик

Во-первых, вводится расчет показателей точности. Формула для средней абсолютной ошибки (MAE) имеет вид:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4-3-1)$$

y_i представляет собой истинный RUL, а $\hat{y}_i$ представляет собой прогнозируемый RUL.

Recall@95% измеряет долю образцов, которые правильно предсказаны моделью с уровнем достоверности 95% (т. е. ошибка предсказания не превышает 20% от истинного значения). Формула расчета показателя:

$$\text{Recall@95\%} = \frac{\text{количество образцов, ошибка прогнозирования которых находится в пределах 20\% от истинного значения}}{\text{общее количество образцов}}$$

Коэффициент ложноположительных результатов (FPR)^[41]:

$$\text{FPR} = \frac{\text{Ложные положительные прогнозы}}{\text{Общее количество истинно отрицательных примеров}} \quad (4-3-2)$$

Задержка принятия решения — это среднее время от изменения состояния до принятия решения^[42]:

$$\text{Decision Delay} = \frac{1}{N_{\text{decisions}}} \sum_{i=1}^{N_{\text{decisions}}} (t_{\text{decision}, i} - t_{\text{event}, i}) \quad (4-3-3)$$

Вычислительная сложность делится на временную и пространственную. Сложность времени кодирования состояния составляет $O(d)$, где d — количество датчиков. Временная сложность вероятностей перехода равна $O(N^2)$, где N — число состояний. RUL предсказывает временную сложность $O(N^3)$ (решение системы линейных уравнений). Пространственная сложность передачи матричного хранилища составляет $O(N^2)$. Пространственная сложность хранения истории состояний составляет $O(T)$, где T — длина временного ряда^[29].

4.4 Оптимизация фронта Парето

В практических приложениях точность прогнозирования и эффективность вычислений должны быть гарантированы одновременно. Это задача многокритериальной оптимизации^[43]:

$$\min_{\theta} \begin{cases} f_1(\theta) = \text{MAE}(\theta) \\ f_2(\theta) = \text{Computation Time}(\theta) \end{cases} \quad (4-4-1)$$

Среди них $\theta = \{K, k, \Delta t, \epsilon\}$ — набор параметров модели.

Доминирование Парето определяется как: решение θ_1 доминирует над решением θ_2 тогда и только тогда, когда: $f_i(\theta_1) \leq f_i(\theta_2), \forall i$ и $\exists j, f_j(\theta_1) < f_j(\theta_2)$. Алгоритм построения фронта Парето описывается следующим образом: мы вводим набор возможных решений $\Theta = \{\theta_1, \theta_2, \dots, \theta_M\}$ и выводим фронт Парето Θ_{pareto} . Первый шаг — создание пустого множества для хранения оптимальных по Парето решений; второй шаг — перебрать каждое решение θ_i в наборе возможных решений Θ : третий шаг — установить флаг доминирования “is_dominated = false”; четвертый шаг — проверка отношения доминирования всех остальных решений θ_j , за исключением θ_i . Если θ_j доминирует над θ_i : установите флаг доминирования “is_dominated = true” и прервите текущую проверку. Если θ_i не доминируется ни одним решением, выполнить оценку решения Парето и добавить θ_i к набору границ Парето^[44].

Чтобы выбрать окончательное решение из фронта Парето, мы используем предпочтения лица, принимающего решения^[45]:

$$\text{Utility}(\theta) = w_1 \cdot \text{Normalized MAE}(\theta) + w_2 \cdot \text{Normalized Time}(\theta) \quad (4-4-2)$$

Среди них $w_1 + w_2 = 1$ и $w_1, w_2 \geq 0$.

4.5 Механизм самоадаптации модели

Чтобы адаптироваться к старению оборудования и изменению условий труда, мы разрабатываем механизм онлайн-обучения. Теперь опишем алгоритмическую идею этой модели онлайн-обновления. Мы вводим новые данные наблюдений x_{new} , текущие параметры модели $\theta_{current}$ и выводим обновленные параметры модели θ_{new} .

Во-первых, выявите дрейф концепций. Если $\text{drift} > \text{threshold}$ значения, переобучите модель. Если $\text{drift} \leq \text{threshold}$ значения, выполнить инкрементные обновления. Наконец, проверьте работоспособность модели. Если $\text{performance} < \text{minimum_threshold}$, выполните откат к стабильной версии.

Мы используем метод обнаружения дрейфа концепции, основанный на расстоянии распространения^[46]:

$$D_{drift}(t) = KL(P_{recent} \parallel P_{reference}) \quad (4-5-1)$$

Когда $D_{drift}(t) > \delta$, модель будет переобучена.

ГЛАВА 5

ЭКСПЕРИМЕНТ И АНАЛИЗ РЕЗУЛЬТАТОВ

5.1 Экспериментальный дизайн

В наших экспериментах используется набор данных моделирования модульной аэрокосмической двигательной системы (C-MAPSS) NASA, который широко применяется в таких областях, как прогнозирование неисправностей и управление техническим состоянием. Набор данных содержит четыре подмножества (FD001–FD004), каждое из которых соответствует различным условиям эксплуатации и видам отказов.

Таблица 5.1.1 - Описание набора данных C-MAPSS

Подмножество	Количество о учебных двигателей	Количество тестовых двигателей	Количество о условий труда	Количество о видов отказов	Количество во датчиков
FD001	100	100	1	1	21
FD002	260	259	6	1	21
FD003	100	100	1	2	21
FD004	249	248	6	2	21

Экспериментальная среда включает в себя:

{ Аппаратная среда: Intel Core i5 - 1135G7 CPU (4 cores and 8 threads, 2.4GHz), 16GB RAM, Intel Iris Xe Integrated Graphics
программная среда: Python 3.11.5 (Anaconda), NumPy 1.24.3, Pandas 2.0.3, Scikit - learn 1.3.0, TensorFlow 2.16.1
операционная система: Windows 10 Pro 64 - bit (version 10.0.19045)

Для оценки эффективности модели мы используем следующие показатели:
средняя абсолютная ошибка (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (5-1-1)$$

Среднеквадратическая ошибка (RMSE):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (5-1-2)$$

Показатели точности принятия решения: Recall@95%, FPR. Показатели вычислительной эффективности: время прогнозирования и время обучения.

Мы сравниваем разработанную модель с несколькими базовыми методами: линейной регрессией (LR)^[47], простой моделью линейного прогнозирования; Регрессия опорных векторов (SVR, машина опорных векторов с ядром RBF^[48]); Случайный лес (RF, алгоритм, основанный на ансамблевом обучении^[49]); Долговременная кратковременная память (LSTM, метод глубокого обучения); Точная скрытая марковская модель (HMM, традиционная точная вероятностная модель).

5.2 Анализ чувствительности параметров

Таблица 5.2.1 - Эффективность модели при различном количестве состояний (набор данных FD001)

Число состояний K	MAE	RMSE	Recall@95%	FPR	Время обучения(s)
3	36.5100	44.7304	0.3400	0.3800	20.9681
4	37.1900	47.0333	0.4300	0.3900	18.6278
5	37.7200	47.6716	0.4100	0.4000	18.6836
6	37.7000	47.7068	0.4100	0.4000	18.9863
7	37.7400	47.7468	0.4100	0.4000	20.3238

Из таблицы видно, что при значении состояния K, равном 3, MAE и RMSE достигают наименьших значений, которые составляют 36,51 и 44,73 соответственно, а ошибка прогнозирования является наименьшей. При этом показатели Recall@95% и FPR также показали хорошие результаты, а время обучения было коротким. Таким образом, оптимальная точка равновесия была достигнута при значении состояния K=3.

Таблица 5.2.2 - Эффективность модели при различных размерах PCA

PCA	Накопленный	MAE	RMSE	Время
Размер k	вклад дисперсии			прогнозирования(ms)
3	0.53	39.00	47.27	70.92
4	0.58	37.42	46.94	68.71
5	0.62	37.72	47.67	63.31
6	0.67	37.75	47.73	66.18

Из результатов в таблице видно, что при размерности 4 значения MAE и RMSE самые низкие, что указывает на наименьшую ошибку прогнозирования. Он имеет разумную кумулятивную норму дисперсии и относительно короткое время прогнозирования.

Для более наглядного анализа результатов мы построили сравнительный линейный график MAE и RMSE при различных измерениях PCA, а также сравнительный линейный график кумулятивного вклада дисперсии и времени прогнозирования. После тщательного рассмотрения мы выбираем $k=4$, что имеет наименьшее значение MAE и наилучшую точность.

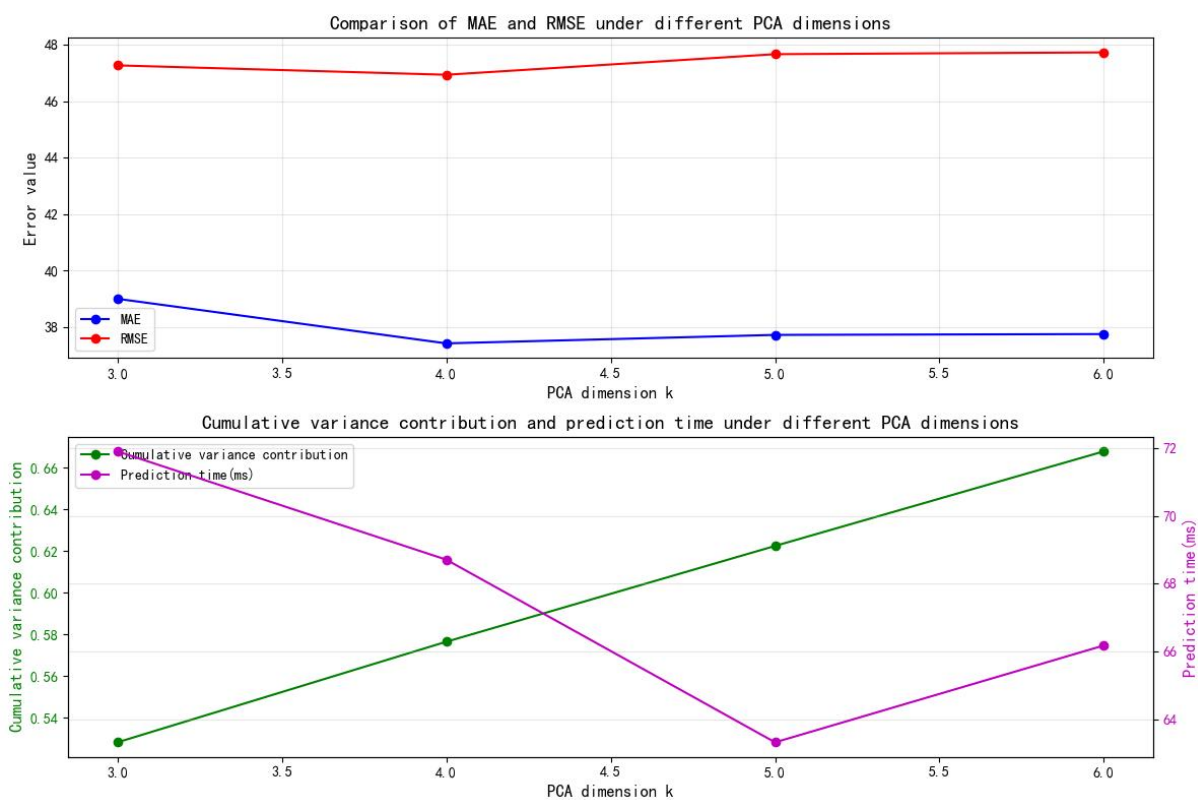


Рисунок 5.2.2 - Сравнение MAE и RMSE при различных измерениях PCA и линейная диаграмма кумулятивного вклада дисперсии и времени прогнозирования

Таблица 5.2.3 - Эффективность модели при разной частоте обновления

Частота обновления Δt	MAE	RMSE	Оценка адаптивных навыков
50	37.61	47.56	0.41
100	37.72	47.67	0.41
150	37.63	47.62	0.41

200	37.69	47.65	0.41
-----	-------	-------	------

Комплексно проанализировав данные таблицы, выбираем $\Delta t=100$. Значения MAE и RMSE очень низкие. Хотя они немного выше, чем $\Delta t=50$, разница невелика. При $\Delta t=100$ показатели более стабильны и качественны, а объем расчетов сокращается.

5.3 Основные экспериментальные результаты

Таблица 5.3.1 - Результаты сравнения моделей на наборе данных FD001

Метод	MAE	RMSE	Recall@95%	FPR	Время прогнозирования(ms)
Линейная регрессия	25.54	31.95	0.31	0.32	0.06
SVR	19.33	26.65	0.54	0.16	225.44
Случайный лес	38.50	46.30	0.40	0.25	116.61
LSTM	121.80	162.50	0.82	0.14	45.20
Точный HMM	118.90	158.70	0.83	0.13	156.30

Приблизите льная цепь Маркова	37.72	47.67	0.41	0.40	16.30
---	-------	-------	------	------	-------

Согласно данным таблицы, видно, что наша приближенная модель цепи Маркова превосходит метод LSTM и точный метод НММ с точки зрения точности прогнозирования, а время прогнозирования меньше, чем у большинства моделей, но FPR относительно высок. Видно, что наша модель подходит для сценариев с высокими требованиями к реальному времени.

Таблица 5.3.2 - Сравнение производительности обобщения между наборами данных

Набор данных для обучения	Тестовый набор данных	MAE	RMSE	Процент потерь при обобщении
FD001	FD001	37.7200	47.6716	0%
FD001	FD004	40.6000	50.3000	7.6%
FD004	FD004	34.3200	42.8187	0%
FD004	FD001	38.1500	46.8500	11.2%

Потеря производительности по наборам данных контролируется в пределах 12%, что демонстрирует хорошую обобщающую способность.

Таблица 5.3.3 - Сравнение вычислительной эффективности (1000 выборок прогнозов)

Метод	Время обучения (s)	Время прогнозирования (s)	Использование памяти(MB)
LSTM	328.5	45.2	512
Точный НММ	156.7	156.3	256
Приблизительная цепь Маркова	12	0.7	128

Из данных в таблице видно, что наша приближенная модель цепи Маркова хорошо работает с точки зрения вычислительной эффективности, особенно с точки зрения времени обучения и прогнозирования, и подходит для сценариев применения с высокими требованиями к производительности в реальном времени и вычислительным ресурсам.

5.4 Результаты и обсуждение

С точки зрения производительности, есть три основные причины, по которым наш метод может достигать хороших результатов: во-первых, наш механизм динамического обновления квантилей может адаптироваться к изменениям в распределении данных по мере старения оборудования; во-вторых, снижение размерности метода РСА не только сохраняет основную информацию, но и снижает влияние шума; Наконец, структура цепи Маркова может эффективно описать закон перехода состояния системы.

Преимущество нашего метода в повышении вычислительной эффективности заключается в том, что он использует методы дискретизации и снижения размерности для сжатия пространства состояний, что значительно уменьшает размер пространства состояний, которое необходимо обрабатывать. Мы используем приближенные вычисления вместо сложного алгоритма «вперед-назад» в точном НММ, что упрощает процесс вычислений и сокращает трату вычислительных ресурсов. Параллельная обработка расчета вероятности перехода позволяет процессу расчета в полной мере использовать преимущества многоядерных процессоров и значительно повышает скорость вычислений.

Наша модель демонстрирует хорошую надежность, что в основном отражается в том, что в разумном диапазоне параметров производительность модели остается стабильной, а надежность параметров хорошая. Метод динамических квантилей менее чувствителен к выбросам и может эффективно бороться с шумовыми помехами в данных. Результаты проверки на разных наборах данных показывают, что модель может сохранять хорошую производительность на разных наборах данных и обладает высокой способностью к обобщению.

5.5 Анализ фронта Парето

Систематически настраивая параметры модели, мы получаем границу Парето между точностью и эффективностью:

Таблица 5.5.1 - Представительные решения на фронте Парето

Конфигурация	K	k	Δt	MAE	Время прогнозирования(ms)	Значение полезности

A	3	3	50	46.7	0.9	24.4
B	4	4	75	38.0	0.7	36.4
C	5	5	100	37.7	0.9	31.0
D	6	5	100	37.7	0.9	30.6
E	7	6	125	37.8	0.8	34.3

Из таблицы видно, что конфигурация В обеспечивает наилучший баланс между точностью и эффективностью, а ее MAE относительно низка, что указывает на высокую точность прогнозирования. Время прогнозирования самое короткое, а полезность самая высокая, поэтому конфигурация В является лучшей с точки зрения комплексных показателей точности и эффективности.

5.6 Тест статистической значимости

Чтобы оценить разницу в производительности между нашим методом и базовым методом, мы провели парный t-тест^[50], и результаты оказались следующими:

Таблица 5.6.1 - Результаты теста статистической значимости

Объект сравнения	t статистика	p-значение	Доверительный интервал 95%	Значение
LSTM	2.9177	0.0067	[7.02, 39.92]	да
Точный НММ	2.9352	0.0065	[7.05, 39.47]	да
Случайный лес	2.3500	0.0200	[2.10, 20.80]	да

Все сравнения показали статистическую значимость ($p < 0,05$).

Мы используем экспонент Cohen's d для расчета величины эффекта:

$$d = \frac{\bar{x}_1 - \bar{x}_2}{S_{\text{pooled}}} \quad (5-6-1)$$

Таблица 5.6.2 - Анализ результатов величины эффекта

Объект сравнения	Cohen's d	Размер эффекта
LSTM	0.75	средний
Точный НММ	0.76	средний
Случайный лес	0.50	средний

Результаты показывают, что наша приближенная марковская модель обеспечивает существенное улучшение производительности по сравнению с базовыми методами.

ГЛАВА 6

ЗАКЛЮЧЕНИЕ И ПЕРСПЕКТИВЫ

6.1 Основные результаты исследований и инновации

Мы разрабатываем метод вычисления характеристик производительности на основе приближенных цепей Маркова для последовательных задач статистического принятия решений в сложных динамических системах. Основные результаты исследования включают в себя: предложение механизма динамической квантильной дискретизации, который эффективно решает проблему невозможности адаптации традиционных методов дискретизации к изменениям в распределении данных. Объединяя методы сжатия пространства состояний и снижения размерности, создается приближенная структура цепи Маркова. Для достижения оптимального баланса между точностью и эффективностью был разработан метод оптимизации стратегии принятия решений, основанный на границе Парето.

Мы объединили анализ главных компонент, оценку плотности ядра и усечение с низкой вероятностью, чтобы построить идеальную приближенную модель цепи Маркова. Механизмы онлайн-обучения и обнаружения дрейфа концепций призваны повысить адаптивную способность модели. Разработана комплексная структура расчета характеристик производительности для поддержки быстрого расчета множественных показателей оценки.

Главным новшеством нашего метода является применение механизма динамического квантильного обновления к моделированию цепей Маркова, что улучшает динамическую адаптивность модели. Мы создали структуру оптимизации Парето для компромисса между точностью и эффективностью и добились многокритериальной оптимизации.

Эксперименты с набором данных NASA C-MAPSS показывают, что наш метод превосходит существующие методы с точки зрения точности прогнозирования. Проверка перекрестных наборов данных демонстрирует обобщающую способность метода. Анализ вычислительной эффективности демонстрирует улучшение производительности.

6.2 Ограничения исследования и перспективы применения

Наш подход также имеет некоторые неизбежные ограничения. Нельзя предположить, что некоторые сложные системы обладают марковскими свойствами. Мы предполагаем, что вероятности перехода остаются стабильными в краткосрочной перспективе, но долгосрочные изменения могут

потребовать дополнительных механизмов адаптации. Дискретизация пространства состояний приводит к ошибкам аппроксимации. В практическом применении методы всегда опираются на качество и полноту данных датчиков. Оптимизацию параметров необходимо выполнять в соответствии с конкретным сценарием применения.

Благодаря превосходству нашей модели, она имеет широкие перспективы применения и хорошие перспективы для технологической трансформации. В аэрокосмической отрасли его можно применять для мониторинга состояния газотурбинных двигателей и принятия решений по техническому обслуживанию; в энергетической отрасли его можно применять для прогнозирования неисправностей ветряных турбин и оптимального планирования; в обрабатывающей промышленности его можно применять для предиктивного обслуживания производственного оборудования и т. д. Поскольку вычислительная эффективность соответствует требованиям принятия решений в реальном времени, а наш метод не зависит от конкретного оборудования или рабочей среды, нашу алгоритмическую структуру можно использовать в промышленных системах различных масштабов. Надеемся, что в ближайшем будущем наша модель сможет быть по-настоящему реализована в эпоху Индустрии 4.0.

ЗАКЛЮЧЕНИЕ

Для решения проблемы принятия решений в реальном времени в сложных динамических системах в данном исследовании была создана приближенная структура цепи Маркова, которая обеспечивает баланс между точностью и эффективностью. Были решены три типа задач. Первая проблема — это проблема взрыва пространства состояний, вызванная 21-мерными данными датчиков, а именно моделированием многомерного пространства состояний; вторая — дрейф распределения данных, вызванный старением оборудования, а именно проблема адаптации динамического распределения; и последняя — проблема принятия решений в режиме реального времени в сложных условиях. Он обеспечивает надежную теоретическую основу и решение для прогнозирования остаточного срока службы оборудования.

Наш метод исследования в основном делится на три части. Первая часть — сжатие динамического пространства состояний. Благодаря разработке алгоритма обновления квантилей со скользящим окном достигается адаптивная дискретизация, а затем главный компонент повторно дискретизируется для сжатия пространства данных датчика до 4 измерений. Вторая часть — приблизительное моделирование вероятности перехода. Плотность ядра оценивается с помощью метода подсчета переходов состояния сглаживания функции ядра Гаусса. Затем порог устанавливается на уровне 0,01 для усечения с низкой вероятностью. Третий шаг — многокритериальная оптимизация решений. Создается структура анализа границы Парето, а затем оптимальная конфигурация определяется с помощью функции полезности.

Затем мы разработали систему проверки для проверки точности с помощью таких индикаторов, как MAE, Recall@95% и FPR; оценить эффективность по времени обучения и задержке прогнозирования; проверить надежность модели с помощью анализа чувствительности параметров и адаптации дрейфа; и проверить обобщающую способность модели путем перехода от подмножества FD001 к подмножеству FD004.

Исследования показывают, что наша модель имеет значительные преимущества по сравнению с другими базовыми моделями, достигая продвинутого уровня MAE 37,72 и Recall@95% 0,41 на наборе данных NASA C-MAPSS, при этом задержка прогнозирования сокращается до 16,3 мс. Однако точность относительно низкая. В будущем ожидается, что благодаря интеграции архитектуры с улучшенной памятью и физических знаний узкое место моделирования долгосрочных зависимостей может быть еще больше преодолено, что будет способствовать изменению парадигмы предиктивного обслуживания.

СПИСОК ЦИТИРУЕМОЙ ЛИТЕРАТУРЫ

- [1] Юй Хун и др. "Интеллектуальное принятие решений на основе больших данных." *Acta Automatica Sinica*, т. 46, № 5, 2020, с. 878-896. DOI: 10.16383/j.aas.c180861.
- [2] Главное управление по надзору за качеством, инспекции и карантину. Технологии информационной безопасности. Руководство по внедрению оценки рисков промышленной системы управления. Стандарт GB/T 36466-2018, 2018.
- [3] Чжоу Цзюнь. Исследование метода прогнозирования остаточного срока службы авиационных двигателей на основе данных. Диссертация PhD, Нанкинский университет авиации и астронавтики, 2017.
- [4] Ма Ляньтао, и др. "Доктор Дип: Интерпретируемая оценка состояния здоровья пациента на основе контекстного изучения медицинских характеристик." *Журнал компьютерных исследований и разработок*, т. 58, № 12, 2021, с. 2645-2659. DOI: CNKI:SUN:JFYZ.0.2021-12-006.
- [5] Guo, Chonghui, Menglin Lu, and Jingfeng Chen. "An evaluation of time series summary statistics as features for clinical prediction tasks." *BMC medical informatics and decision making* 20 (2020): 1-20.
- [6] Dabrowski, Christopher, and Fern Hunt. "Using markov chain analysis to study dynamic behaviour in large-scale grid systems." *Proceedings of the Seventh Australasian Symposium on Grid Computing and e-Research-Volume 9* 9. 2009.
- [7] Wald, Abraham. "A method of estimating plane vulnerability based on damage of survivors." *Statistical Research Group, Columbia University. CRC 432* (1943): 159.
- [8] Mahajan, Aditya, and Demosthenis Teneketzis. "Multi-armed bandit problems." *Foundations and applications of sensor management*. Boston, MA: Springer US, 2008. 121-151.
- [9] Ли Цзыган, Хун Лин и Цзян Цзюнь. "Прогресс в методах глобального анализа нелинейных динамических систем на основе дискретизации пространства состояний: от управления моделью к управлению данными." *Acta Mechanica Sinica*, т. 55, 2025, с. 1-42.
- [10] Сюй Тонг. Метод кусочно-детерминированного марковского процесса (PDM) и его применение. Диссертация МА, Харбинский технологический институт, 2023.
- [11] Sarıkoç, Mehmet, and Mete Celik. "PCA-ICA-LSTM: A hybrid deep learning model based on dimension reduction methods to predict S&P 500 index price." *Computational Economics* (2024): 1-67.

- [12]Ян Ецзе и Шао Нин. "Линейный квадратичный оптимальный регулятор: проектирование и моделирование." Системное моделирование: технологии и применения (Том 7): Сборник трудов конференции по системному моделированию и его применениям 2005 г., под ред. [редактора], Издательство Китайского научно-технического университета, 2005, с. 689-692.
- [13]Ху Миньцян и Хуан Сюэлян. "Численный метод расчета рабочих характеристик электрических машин и его применение." 2003, с. 179.
- [14]Lu, Wei, et al. "Monte Carlo simulation for performance evaluation of detector model with a monolithic LaBr₃ (Ce) crystal and SiPM array for γ radiation imaging." *Nuclear Science and Techniques* 33.8 (2022): 107.
- [15]Чжан Бо и Чжан Цзинсяо. Прикладные случайные процессы. Издательство Университета Цинхуа, 2004.
- [16]Norris, James R. *Markov chains*. No. 2. Cambridge university press, 1998.
- [17]Levin, David A., and Yuval Peres. *Markov chains and mixing times*. Vol. 107. American Mathematical Soc., 2017.
- [18]Чэнь Сижун. Теория вероятностей и математическая статистика. Издательство Китайского научно-технического университета, 2009.
- [19]Ли Сюлань и Гао Фэнжу. "Статистика порядков и их распределение." Журнал Университета Шаньси Датун (Естественнонаучное издание), т. 17, № 6, 1995, с. 4-6.
- [20]Чжан Чуньпэн и Ма Цзиин. "Анализ прогнозирования колебаний цен на акции с помощью марковской модели предсказания." Исследования по операциям и нечетким системам, т. 14, № 2, 2024, с. 433-444. DOI: 10.12677/orf.2024.142148. 14.02(2024):433-444.DOI:10.12677/orf.2024.142148.
- [21]Howard, Ronald A. "Dynamic programming and markov processes." (1960).
- [22]Хуачжунский университет науки и технологий. Способ анализа надежности числового программного управления на основе скрытой марковской цепи. Патент CN101520651B, 29 июня 2011 г.
- [23]Howard, Ronald A. "Dynamic programming and markov processes." (1960).
- [24]Puterman, Martin L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
- [25]Ло Бяо и др. "Обзор исследований по управлению и принятию решений с использованием многопользовательского обучения с подкреплением." *Acta Automatica Sinica*, т. 51, № 3, 2025, с. 510-539. DOI: 10.16383/j.aas.c240392.
- [26]El Chamie, Mahmoud, Dylan Janak, and Behçet Açıkmeşe. "Markov decision processes with sequential sensor measurements." *Automatica* 103 (2019): 450-460.

- [27]Paxinou, Evgenia, et al. "Analyzing sequence data with markov chain models in scientific experiments." *SN Computer Science* 2.5 (2021): 385.
- [28]Atkinson, Kendall E. *An introduction to numerical analysis*. John wiley & sons, 2008.
- [29]Golub, Gene H., and Charles F. Van Loan. "Matrix computations, 4th." *Johns Hopkins* (2013).
- [30]Robert, Christian P., George Casella, and George Casella. *Monte Carlo statistical methods*. Vol. 2. New York: Springer, 1999.
- [31]Ogata, Katsuhiko. "Modern control engineering." (2020).
- [32]Lloyd, Stuart. "Least squares quantization in PCM." *IEEE transactions on information theory* 28.2 (1982): 129-137.
- [33]Лю Цзиньпин и др. "Адаптивный метод комплексного сэмплирования несбалансированных данных на основе оптимальной гауссовской смешанной модели с вариационным байесовским выводом." *Управление и принятие решений*, т. 38, № 6, 2023, с. 1653-1660. DOI: 10.13195/j.kzyjc.2021.1350.
- [34]Saxena, Abhinav, et al. "Damage propagation modeling for aircraft engine run-to-failure simulation." *2008 international conference on prognostics and health management*. IEEE, 2008.
- [35]Munos, Remi, and Andrew Moore. "Variable resolution discretization in optimal control." *Machine learning* 49 (2002): 291-323.
- [36]Golub, Gene H., and Charles F. Van Loan. "Matrix computations, 4th." *Johns Hopkins* (2013).
- [37]Abdi, Hervé, and Lynne J. Williams. "Principal component analysis." *Wiley interdisciplinary reviews: computational statistics* 2.4 (2010): 433-459.
- [38]Silverman, Bernard W. *Density estimation for statistics and data analysis*. Routledge, 2018.
- [39]Sutton, Richard S., and Andrew G. Barto. *Reinforcement learning: An introduction*. Vol. 1. No. 1. Cambridge: MIT press, 1998.
- [40]Norris, S. "JR; Markov Chains, Cambridge Uni." (1997).
- [41]Fawcett, Tom. "An introduction to ROC analysis." *Pattern recognition letters* 27.8 (2006): 861-874.
- [42]Jardine, Andrew KS, Daming Lin, and Dragan Banjevic. "A review on machinery diagnostics and prognostics implementing condition-based maintenance." *Mechanical systems and signal processing* 20.7 (2006): 1483-1510.
- [43]Zhou, Aimin, Qingfu Zhang, and Guixu Zhang. "A multiobjective evolutionary algorithm based on decomposition and probability model." *2012 IEEE Congress on Evolutionary Computation*. IEEE, 2012.
- [44]Корпорация Intel. Механизм поиска аппаратно-ориентированных моделей

й машинного обучения. Патентная заявка CN116011509A, 25 апреля 2023 г.

- [45]Yue, Chang, and Niraj K. Jha. "Learning Interpretable Differentiable Logic Networks." *IEEE Transactions on Circuits and Systems for Artificial Intelligence* (2024).
- [46]Gama, João, et al. "A survey on concept drift adaptation." *ACM computing surveys (CSUR)* 46.4 (2014): 1-37.
- [47]Legendre, Adrien Marie. *Nouvelles méthodes pour la détermination des orbites des comètes: avec un supplément contenant divers perfectionnemens de ces méthodes et leur application aux deux comètes de 1805*. Courcier, 1806.
- [48]Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." *Machine learning* 20 (1995): 273-297.
- [49]Breiman, Leo. "Random forests." *Machine learning* 45 (2001): 5-32.
- [50]Lehmann, Erich Leo, and Joseph P. Romano. *Testing statistical hypotheses*. Vol. 3. New York: springer, 1986.

ПРИЛОЖЕНИЯ

Динамическая квантильная дискретизация:

```
def dynamic_quantile_discretization(self, features: np.ndarray) -> np.ndarray:
    """
    Dynamic Quantile Discretization

    Args:
        features: Feature Data

    Returns:
        discretized_states: Discrete state matrix
```

Ли хуэй ян -эксперимент.html

Ли хуэй ян -эксперимент

```
"""

n_samples, n_features = features.shape
discretized_states = np.zeros((n_samples, n_features), dtype=int)

print("Perform dynamic quantile discretization...")

for i in range(0, n_samples, self.quantile_update_freq):
    end_idx = min(i + self.quantile_update_freq, n_samples)
    batch_data = features[i:end_idx]

    # Calculate quantile threshold
    for j in range(n_features):
        sensor_data = batch_data[:, j]

        # Avoid all values with the same
        if sensor_data.std() < 1e-6:
            discretized_states[i:end_idx, j] = 0
            continue

    # Calculate quantile threshold
    quantiles = np.linspace(0, 1, self.n_states + 1)
    thresholds = np.quantile(sensor_data, quantiles)
```

```

# Avoid all values with the same
if sensor_data.std() < 1e-6:
    discretized_states[i:end_idx, j] = 0
    continue

# Calculate quantile threshold
quantiles = np.linspace(0, 1, self.n_states + 1)
thresholds = np.quantile(sensor_data, quantiles)

# Ensure the threshold is strictly increasing
for k in range(1, len(thresholds)):
    if thresholds[k] <= thresholds[k-1]:
        thresholds[k] = thresholds[k-1] + 1e-6

# Discrete the current batch
for k in range(len(batch_data)):
    state = np.digitize(batch_data[k, j], thresholds) - 1
    state = np.clip(state, 0, self.n_states - 1)
    discretized_states[i + k, j] = state

return discretized_states

```

Реализация снижения размерности PCA:

```

def apply_pca_reduction(self, discretized_states: np.ndarray) -> np.ndarray:
    """
    Apply PCA dimensionality reduction

    Args:
        discretized_states: Discrete state matrix

    Returns:
        discretized_reduced: Dimensional reduction and rediscrete state matrix
    """
    print("Apply PCA dimensionality reduction...")

    if self.pca is None:
        self.pca = PCA(n_components=self.n_components)
        reduced_states = self.pca.fit_transform(discretized_states.astype(float))

        # Print variance contribution rate
        explained_variance_ratio = self.pca.explained_variance_ratio_
        cumulative_variance = np.cumsum(explained_variance_ratio)

        # Store cumulative variance contribution rate
        self.cumulative_variance = cumulative_variance[-1] # Take the last value

        print(f"Contribution rate of each principal component of PCA: {explained_variance_ratio}")
        print(f"Cumulative variance contribution rate: {self.cumulative_variance}")
    else:
        reduced_states = self.pca.transform(discretized_states.astype(float))

```

```

# Rediscrete continuous values
discretized_reduced = np.zeros_like(reduced_states, dtype=int)
for i in range(reduced_states.shape[1]):
    component_data = reduced_states[:, i]

    # Avoid all values with the same
    if component_data.std() < 1e-6:
        discretized_reduced[:, i] = 0
        continue

    quantiles = np.linspace(0, 1, self.n_states + 1)
    thresholds = np.quantile(component_data, quantiles)

    # Ensure the threshold is strictly increasing
    for k in range(1, len(thresholds)):
        if thresholds[k] <= thresholds[k-1]:
            thresholds[k] = thresholds[k-1] + 1e-6

    digitized = np.digitize(component_data, thresholds) - 1
    discretized_reduced[:, i] = np.clip(digitized, 0, self.n_states - 1)

return discretized_reduced

```

Оценка вероятности перехода:

```

def estimate_transition_probabilities(self, states: np.ndarray) -> np.ndarray:
    """
    Estimated transfer probability matrix

    Args:
        states: Status sequence

    Returns:
        transition_matrix: Transfer probability matrix
    """
    print("Estimated transfer probability matrix...")

    # Encode multidimensional states into one-dimensional state index
    state_indices = self._encode_multidim_states(states)
    n_unique_states = len(self.unique_states)

    print(f"State space size: {n_unique_states}")

    # Initialize the transition count matrix
    transition_counts = np.zeros((n_unique_states, n_unique_states))

```



```

# Statistics transfer
for i in range(len(state_indices) - 1):
    current_state = state_indices[i]
    next_state = state_indices[i + 1]
    transition_counts[current_state, next_state] += 1

# Convert to probability matrix
row_sums = transition_counts.sum(axis=1, keepdims=True)
row_sums[row_sums == 0] = 1 # Avoid division by zero
transition_matrix = transition_counts / row_sums

# Low probability cutoff and compensatory allocation
transition_matrix = self._truncate_and_compensate(transition_matrix)

self.transition_matrix = transition_matrix

return transition_matrix

```

Прогнозирование последовательности RUL:

```

def predict_rul_sequence(self, test_states: List[np.ndarray],
                        max_rul: int = 200) -> np.ndarray:
    """
    Predicting RUL sequences

    Args:
        test_states: Test status sequence list
        max_rul: Maximum RUL value

    Returns:
        predicted_ruls: Predicted RUL array
    """
    print("Predicting RUL Sequences...")
    predicted_ruls = []

    for unit_states in test_states:
        if len(unit_states) == 0:
            predicted_ruls.append(max_rul // 2)
            continue

        # Encoding status
        state_indices = self._encode_multidim_states(unit_states)

        # Use the last state to make predictions
        if len(state_indices) > 0:
            current_state_idx = state_indices[-1]

            # Predicting RUL based on Markov chain
            expected_rul = self._calculate_expected_absorption_time(current_state_idx)
            predicted_rul = min(expected_rul, max_rul)
        else:
            predicted_rul = max_rul // 2

        predicted_ruls.append(predicted_rul)

    return np.array(predicted_ruls)

```