

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**ПРИМЕНЕНИЕ МЕТОДОВ ДЕКОМПОЗИЦИИ МОД И ПОЛОСОВОЙ
ФИЛЬТРАЦИИ СИГНАЛА ДЛЯ ИЗУЧЕНИЯ СВОЙСТВ
ВРЕМЕННЫХ РЯДОВ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Гизатуллина Данилы Александровича

Научный руководитель:

Д. Т. Н.

_____ А.С.Фалькович

подпись, дата

Зав. кафедрой информатики и

программирования, к. ф.-м. н., доцент: _____

М.В.Огнева

подпись, дата

Саратов 2019

ВВЕДЕНИЕ

Актуальность темы.

Временные ряды представляют собой статистический материал, содержащий набор значений какой-либо величины в различные моменты времени. Анализ временных рядов широко применяется в различных науках: от биологии до физики. Выявление структуры временного ряда в ходе анализа позволяет строить достоверные математические модели изучаемого явления и получать соответствующие предсказания на их основе, тем самым выполняя одну из главных задач прикладной науки.

Цель бакалаврской работы – разработка программы для анализа временных рядов методом декомпозиции мод (методом Гильберта-Хуанга) и полосовой фильтрации сигнала применительно к изучению свойств временных рядов.

Поставленная цель определила **следующие задачи:**

1. Изучить и проанализировать свойства временных рядов и методы их анализа;
2. Изучить методы декомпозиции мод и полосовой фильтрации;
3. Составить программу на языке C#, реализующую методы декомпозиции мод и полосовой фильтрации;
4. Протестировать программу на числовом примере, содержащем данные о ценах на зерно и примере, содержащем данные о поставках газа потребителю за полтора года

Методологические основы применения методов декомпозиции мод и полосовой фильтрации сигнала для изучения свойств временных рядов представлены в работах Б.П. Безручко, Н.А. Ховановой, Л.Р. Рабинера, Н. Хуанга, Н.С. Бахвалова

Структура и объём работы. Бакалаврская работа состоит из введения, 2-х разделов, заключения, списка использованных источников и 8

приложений. Общий объем работы – 63 страниц, из них 47 страниц – основное содержание, включая 22 рисунков и 2 таблиц, цифровой носитель в качестве приложения, список использованных источников информации – 22 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Эмпирическая декомпозиция мод и метод полосовой фильтрации сигналов как способы анализа временных рядов» посвящен теоретическому описанию основных особенностей представленных в работе методов.

Во многих ситуациях статистические данные об исследуемом объекте представляют собой конечную последовательность наблюдаемой величины в различные моменты времени т. е. временной ряд:

$$\eta(t_1), \eta(t_2), \eta(t_1) \dots, \eta(t_N).$$

где

$$t_1, t_2, t_1 \dots, t_N$$

моменты наблюдений. N-длина ряда. Временные ряды широко применяются при измерении технических, химических, биологических, физических и т.д. процессов

Основной теоретической работой является описание метода декомпозиции эмпирических мод.

Метод основан на декомпозиции конечного, и зачастую небольшого числа, так называемых «внутренних модовых функции» или просто «мод». В результате преобразования методом исходный сигнал представляется в виде суммы конечного числа компонент (эмпирических мод) и некоторого

остатка, как правило, близкого к нулю. т.е. 5 разложение имеет следующий вид:

$$X(t) = \sum_{j=1}^n c_j + r_n$$

И описания используемого в работе алгоритма декомпозиции эмпирических мод:

На вход алгоритма подается некоторый сигнал или наблюдаемый процесс $X(t)$. В частном случае это дискретный сигнал, представленный в виде эквидистантного временного ряда. Опишем процесс просеивания сигнала:

1. Определяются все локальные экстремумы сигнала $X(t)$. Математически сигнал уже может является эмпирической модой. В таком случае метод останавливается и разложения сигнала не происходит. На практике такой случай при наблюдении физического процесса невозможен, а разложение такого сигнала является бессмысленным. Как правило предполагается, что исходный временной ряд имеет как минимум один локальный максимум и один локальный минимум.

2. Получение верхней и нижней огибающей (в дальнейшем $e_{\max}(t)$ и $e_{\min}(t)$ соответственно) сигнала методом интерполяции полученных локальных экстремумов (локальных максимумов и минимумов соответственно). Допускается применение линейной интерполяции, хотя часто отмечается, что лучшим вариантом будет интерполяции кубическими сплайнами. Используемые в работе методы интерполяции будут рассмотрены ниже.

3. Построение среднего значения $m(t)$ между полученными верхними и нижними огибающими: $m(t) = (e_{\max}(t) + e_{\min}(t))/2$.

4. Получение компоненты $d(t) = X(t) - m(t)$.

5. Если полученная компонента $d(t)$ удовлетворяет определению эмпирической моды, то процедура просеивания останавливается. Из исходного сигнала вычитается полученная эмпирическая мода: $Y(t) = X(t) - d(t)$. Полученная эмпирическая мода $d(t)$ содержит компоненту исходного сигнала с кратчайшим периодом. Временной ряд $Y(t)$ будет содержать информацию о компонентах с более длительным периодом. Процедура просеивания повторяется, пока разлагаемый временной ряд не станет монотонной функцией (т.е. не будет содержать точек локального экстремума) или станет всюду в своих точках меньше некоторого заранее определенного значения.

Далее описываются общие теоретические положения по распараллеливанию метода декомпозиции эмпирических мод.

Несмотря на то, что каждый этап декомпозиции очередной эмпирической моды зависит от предыдущего, эффективность метода можно попытаться повысить, распараллелив вычисления на каждом отдельном этапе. Так, например, получение среднего значения между огибающими в 3-м шаге алгоритма в случае работы с некоторыми массивами может быть распараллелено, так как каждый элемент конечного результата не зависит от 8 других элементов результата, а только от значений верхней и нижней огибающих. В таком случае, вместо двух последовательно работающих задач по построению интерполянты, получаем две параллельно выполняющихся задачи.

Последующие подразделы посвящены описанию используемых в работе методов интерполяции сигнала:

Метод интерполяции Ньютона основан на построении интерполяционного многочлена, позволяющего вычислить приблизительные значения функции в точках, которые не использовались в узлах интерполяции. В ходе выполнения работы была реализована интерполяционная формула Ньютона с разделенными разностями. Разделенные разности представляют собой обобщение понятия производной,

удобное для применения при работе с дискретными данными, подаваемыми на вход алгоритмов интерполяции в компьютерных системах.

Интерполяционные формулы Лагранжа, Ньютона и Стирлинга и др. при использовании большого числа узлов интерполяции на всем отрезке $[a, b]$ часто приводят к плохому приближению из-за накопления погрешностей в процессе вычислений. Кроме того, из-за расходимости процесса интерполяции увеличение числа узлов не обязательно приводит к повышению точности. Для снижения погрешностей весь отрезок $[a, b]$ разбивается на частичные отрезки и на каждом из них функцию заменяют приближенно полиномом невысокой степени. Это называется кусочно-полиномиальной интерполяцией. Один из способов интерполирования на всем отрезке $[a, b]$ является интерполирование сплайнами. Сплайном называется кусочно-полиномиальная функция, определенная на отрезке $[a, b]$ и имеющая на этом отрезке некоторое количество непрерывных производных. Преимущества интерполяции сплайнами по сравнению с обычными методами интерполяции – в сходимости и устойчивости вычислительного процесса. Сплайн является кубическим многочленом на каждом отрезке между точками интерполяции, непрерывной на всем отрезке $[a, b]$, и совпадающий со значениями интерполируемой функции в точках интерполяции

Следующий подраздел «Полосовая фильтрация сигналов на основе последовательного скользящего усреднения» посвящен описанию работы метода полосовой фильтрации:

Полосовая фильтрация на основе последовательного скользящего усреднения алгоритм выделения средне-масштабной составляющей сигнала, уменьшающий как более быстрые, так и более медленные его компоненты

Предложенный вариант алгоритма полосовой фильтрации требует указания как минимум одного дополнительного параметра фильтрации, а именно окна 13 скользящего сглаживания w . Это, хотя и добавляет некоторую вариативность природе самого метода, все же требует некоторой

дополнительной определенности относительно того, какой результат мы хотим получить. Заметим еще раз, что декомпозиция эмпирических мод для выполнения нуждается только в исходном сигнале. Кроме того, метод декомпозиции эмпирических мод в том виде, в котором он описан выше, позволяет нам получить сразу целое множество этапов фильтрации сигнала, тогда как описанная полосовая фильтрация для этого потребует явного рекурсивного выполнения.

Второй раздел «Реализация метода декомпозиции эмпирических мод на языке C#» посвящен реализации я метода декомпозиции эмпирических мод.

Второй раздел начинается с основания выбора языка C# для реализации описанных выше алгоритмов. Приводится ряд достоинств этого языка программирования, такие как: возможность быстро и легко производить вывод данных пользователю, наличие удобных решений для чтения информации из Excel файлов .xls, богатый выбор среди имеющихся библиотек численных методов и визуализации данных, интеграция с программной платформой .NET framework, поддержка элементов функционального программирования, возможность написания компактных программных решений

Описание проделанной практической работы начинается с описания структуры программы. Приводятся два варианта реализованного метода декомпозиции эмпирических мод: в виде статического класса и в виде не статического класса. Различия между реализациями минимальны, поэтому в дальнейшем рассматривается статическая реализация метода.

Сам метод реализован в виде библиотеки классов .NET. Библиотека включает в себя:

- Класс Signal Element, представляющий собой элемент подающегося на 16 вход метода сигнала
- Статическую и динамическую реализацию метода декомпозиции эмпирических мод;

- Интерфейс Interpolation, предназначенный для работы метода с алгоритмом интерполяции;
- Реализацию интерфейса Interpolation в виде метода интерполяции Ньютона с разделенными разностями;
- Реализацию интерфейса Interpolation для работы с методом интерполяции кубическими сплайнами на базе библиотеки algib;
- Параллельную реализацию метода декомпозиции мод.
- Реализацию метода полосовой фильтрации

Далее в работе подробно расписываются практические аспекты реализации вышеописанных программных объектов. В том числе описание реализации самого метода декомпозиции мод и описание реализованных методов интерполяции. Также описывается параллельная версия метода декомпозиции мод.

После этого описываются реализованные в работе методы ввода и вывода данных для алгоритма. Так, например для ввода данных используется библиотека ExcelDataReader. Для получения данных в нужном для алгоритма формате был создан специальный программный компонент Signal Data Reader. ExcelDataReader является легкой, быстрой и свободно распространяемой библиотекой C# для чтения данных файлов Microsoft Excel ('97-2007). Для работы она не требует дополнительных программ или настроек решения

Вывод результатов работы алгоритма осуществляется через компоненты Windows Forms на основе библиотеки OxyPlot. OxyPlot содержит массу возможностей по графическому отображению данных с широким спектром графиков разного вида и размерности. Помимо Windows Forms OxyPlot доступен для WPF и мобильных приложений. Существенным недостатком библиотеки является неполная документация. Также прилагается описание перевода сигнала из формата, подающегося на вход алгоритма, в формат доступный для OxyPlot.

В дальнейшем практические моменты реализации метода полосовой фильтрации. В работе она представлена классом, содержащим два статических метода `public static IEnumerable Filter (IEnumerable signal, double wStart = 56, double wEnd = 112)` и `private static SignalElement BuildMiddleElement(int i, double w, List signal)`. Метод `Filter` осуществляет основное выполнения алгоритма. На вход он принимает один обязательный параметр – фильтруемый сигнал и два параметра, которые указывать не обязательно.

Метод возвращает вычитание `slow` из `middle`, возвращая в него вычитанную ранее среднюю сигнала `meanValue`.

После всех практических описаний приводятся примеры работы обоих методов, а также сравнение времени выполнения последовательной и параллельной версий метода декомпозиции мод.

ЗАКЛЮЧЕНИЕ

В ходе работы были изучены базовые аспекты анализа сигналов и временных рядов. Изучен и реализован методы декомпозиции эмпирических мод. Изучен и реализован метод полосовой фильтрации сигнала.

Реализован метод интерполяции Ньютона. Изучены библиотеки `algib` и `охугraph` для C#. Составлена программа на языке C#, реализующая метод декомпозиции мод и полосовой фильтрации.

Программа протестирована на числовом примере, содержащем данные о ценах на зерно и примере, содержащем данные о поставках газа потребителю за полтора года.

Для анализа работы метода декомпозиции мод реализована параллельная версия метода. Выполнено сравнение последовательной и параллельной версий метода по времени выполнения.

Все поставленные в работе задачи выполнены, цель работы достигнута.

Основные источники информации:

1. Б.П. Безручко. Математическое моделирование и хаотические временные ряды / Б.П. Безручко, Д.А. Смирнов. — Саратов: Издательство ГосУНЦ "Колледж 2005
2. Н.А. Хованова. Методы анализа временных рядов / Н.А. Хованова, И.А. Хованов. — Саратов: Издательство ГосУНЦ "Колледж 2001.
3. Л.Р. Рабинер,. Теория и применение цифровой обработки сигналов / Л.Р. Рабинер, Б. Гоулд. — Москва: Мир, 1978.
4. Huang, N. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis / N. Huang, Z. Shen, S. Long // Proc. R. Soc. Lond. A. — 1998. — Vol. 454. — Pp. 903–995.
5. Н.С. Бахвалов. Численные методы / Н.С. Бахвалов. — М.: БИНОМ. Лаборатория знаний, 2004. — С. 636.