**ДИПЛОМНА РОБОТА**

**за першим (бакалаврським) рівнем вищої освіти**

Тема: **«Прогнозування первинної інвалідності в Україні з використанням методів регресійного аналізу»**

Виконав: студент 4 курсу, групи ПМ-13-1

напряму підготовки

6.040301 Прикладна математика

(шифр і назва напряму підготовки, спеціальності)

Кривоносов Олександр Дмитрович

(прізвище та ініціали)

Керівник к.т.н., доц., доц. каф. МЗ ЕОМ

(наук. ступ., вчене звання, посада, прізвище та ініціали)

Кузьмено В.І.

(підпис)

Рецензент д.ф.-м.н., доц., проф. каф. ОМ та МК

(наук. ступ., вчене звання, посада, прізвище та ініціали)

Гук Н.А.

(підпис)

ДНІПРОПЕТРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ім. О. ГОНЧАРА

Факультет *прикладної математики*

Кафедра\_\_\_\_*ОМ та МК\_*

Рівень (освітньо-кваліфікаційний рівень) *перший (бакалаврський)*

Напрям підготовки \_\_\_\_\_\_\_\_\_\_\_*6.040301 — Прикладна математика*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Спеціальність\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЗАТВЕРДЖУЮ

Зав. кафедрою ОМ та МК

\_\_\_\_Турчина В.А.\_\_\_\_

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2017 р.

ЗАВДАННЯ

на дипломну роботу студентові

Кривоносову Олександру Дмитровичу

(прізвище, ім’я по батькові)

1. Тема роботи

*Прогнозування первинної інвалідності в Україні з використанням методів регресійного аналізу*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

затверджена наказом по університету від «\_29\_»\_\_\_березня\_\_\_\_\_\_\_\_2017 р. №\_\_464с\_\_

2. Термін здачі студентом закінченої роботи \_\_\_\_05\_\_ червня 2017 р.\_\_\_\_\_\_\_\_

3. Вхідні дані до роботи

*Текстові файли з багатовимірними даними — часовими рядами*

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) *1.Огляд задачі прогнозування часових рядів. 2.Розглянути методи, алгоритми прогнозування. 3.Розробити програмне забезпечення, у якому реалізувати ці методи, алгоритми; провести його тестування. 4. Провести обчислювальні експерименти.*

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

*Графічне зображення початкового часового ряду, та його прогнозування. Скріншоти роботи програми, презентація у Microsoft PowerPoint.*

6. Консультанти по роботі, Із зазначенням розділів проекту, що стосуються їх

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Консультант | Підпис. дата | |
| завдання видав | завдання прийняв |
| Розділ 1 | Кузьменко В.І. |  |  |
| Розділ 2 | Кузьменко В.І. |  |  |
| Розділ 3 | Кузьменко В.І. |  |  |
|  |  |  |  |
|  |  |  |  |

7. Дата видачі завдання 31.03.2017

КАЛЕНДАРНИЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Пор. № | Назва станiв дипломної роботи | Термiн виконання станiв роботи | Примітка |
| 1 | Огляд задачі прогнозування часових рядів та методами її розв’язання, опрацювання літературних джерел | 31.03.2017 |  |
| 2 | Розробка алгоритму прогнозування часових рядів | 15.04.2017 |  |
| 3 | Розробка програмного продукту, що реалізує алгоритм прогнозування | 20.04.2017 |  |
| 4 | Тестування програмного продукту | 11.05.2017 |  |
| 5 | Проведення обчислювальних експериментів | 25.05.2017 |  |
| 6 | Оформлення пояснювальної записки | 01.06.2017 |  |
| 7 | Представлення випускної роботи на кафедру | 12.06.2017 |  |
| 8 | Захист роботи в ДЕК | 16.06.2017 |  |

**Студент** Кривоносов О.Д.

(підпис) (прізвище та ініціали)

**Керівник роботи**  \_\_Кузьменко В.І.

(підпис) (прізвище та ініціали)

# РЕФЕРАТ

*Дипломна робота* «Прогнозування первинної інвалідності в Україні з використанням методів регресійного аналізу» *:  
\_\_ сторінок, \_\_ рисунків, \_\_ таблиць, \_\_ джерел.*

***Об’єктом дослідження*** є задача прогнозування часових рядів.

***Мета роботи:*** розробка алгоритму розв’язання задачі прогнозування часових рядів на основі штучної нейронної мережі для прогнозування первинної інвалідності.

***Методи дослідження***: методи апроксимації, методи оптимізації, методи регресійного аналізу.

***У процесі роботи*** реалізовані багатошаровий перцептрон для апроксимації функцій, гамма-згортки рядів; була створена програма для прогнозування і дослідження якості прогнозування часових рядів, вивчені моделі і методи розв’язування задач прогнозування часових рядів; програма випробувана на даних первинної інвалідності в Україні, даних силу вітру, даних щоденної кількості продаж на торгових точках.

***В результаті роботи*** розроблено програмний продукт, призначений для розв’язання задач прогнозування часових рядів. Алгоритм створений на мові **С++**, інтерфейс користувача створений на мові **R** з використанням програмного пакету **Shiny**.

***Ключові слова:*** ЧАСОВИЙ РЯД, ПРОГНОЗУВАННЯ, АПРОКСИМАЦІЯ ФУНКЦІЇ, ШТУЧНА НЕЙРОННА МЕРЕЖА, ПЕРЦЕПТРОН, ГАММА-ПАМ’ЯТЬ.

# ABSTRACT

**ЗМІСТ**

Вступ

Постановка задачі

Розділ 1. Прогнозування часових рядів

**1.1.** Огляд методів прогнозування часових рядів

**1.2.** Регресійні методи прогнозування часових рядів

**1.3.** Нейронні мережі в прогнозуванні часових рядів

Розділ 2. Програмне забезпечення прогнозування часових рядів

**2.1.** Функціональні можливості та структура програми

**2.2.** Організація обчислювального процесу

**2.3.** Інструкція користувача

Розділ 3. Результати обчислювальних експериментів

Висновки

Список використаної літератури

Список використаної літератури

# ВСТУП

Для вивчення властивостей складних систем широко

використовується підхід, заснований на аналізі сигналів, вироблених

системою. Це дуже актуально в тих випадках, коли математично описати

досліджуваний процес неможливо, але в нашому розпорядженні може бути

деяка характерна спостережувана величина. Тому аналіз систем, особливо

при експериментальних дослідженнях, часто реалізується за допомогою

оброблення реєстрованих сигналів. Наприклад, в медицині — кардіограми, в

сейсмології — коливання земної кори, в метеорології — дані

метеоспостережень.

Часовий ряд - ряд значень будь-яких параметрів досліджуваного

процесу за рівні проміжки часу.

Скалярним часовий рядом називається масив з чисел, що

представляють собою значення деякої змінної , що спостерігається з

деяким постійним кроком по часу, ; . У

аналізі часових рядів виділяються дві основні задачі: задача ідентифікації та

задача прогнозу.

Задача ідентифікації при аналізі часового ряду передбачає відповідь на

питання, які є параметри системи, що породила часовий ряд:

розмірність вкладення, ентропія (перетворення) та інші. Розмірність

вкладення — це мінімальне число динамічних змінних, що однозначно

описують спостережуваний процес. Поняття ентропії пов'язане з

передбачуваністю значень ряду і всієї системи.

Задача прогнозу має на меті за даними спостережень передбачити

майбутні значення вимірюваних характеристик досліджуваного об'єкта, тобто

побудувати прогноз на певний відрізок часу вперед. Є два основні класи

методів прогнозу: локальні і глобальні. Такий поділ проводиться по області

визначення параметрів апроксимуючої функції, що рекурентно встановлює

наступне значення часового ряду за кількома попередніми.

Історично першими були розроблені глобальні методи, в яких на основі

статистичного аналізу пропонувалося використовувати авторегресію, ковзне

середнє і інші. Пізніше в рамках нелінійної динаміки були розроблені нові

практичні методики:

• сингулярний спектральний аналіз (**SSA**), який є глобальним методом;

• локальна апроксимація (**LA**);

• поєднання **SSA-LA**.

Дослідження часових рядів базується на ідеї, що прогнозувати ряд

можна, якщо замість змінних, що входять у вихідну систему, використовувати

так звані вектори затримок спостережень . Є два

варіанти того, як можна подати подати затримку спостережень на вхід до

апроксиматора:

* Використовуючи неявне представлення. Час представляється еффектом,

який він справляє на обробку сигналу, тобто неявним чином. Можна

застосувати згортку до вектора затримок спостережень і отримти одне

число, яке і вважати параметром. Таким чином еффект, який справляе

час на сигнал можна контролювати змінюючи функцію згортки вектора.

* Використовуючи явне представлення. Час має власне конкретне

представлення. Наприклад, система ехолокації кажана посилає

короткий частотно-модульований сигнал, при цьому встановлюючи

єдиний рівень інтенсивності для кожного з частотних каналів на

короткий період FM-розгортки. Для того щоб отримати точну

інформацію про відстань до цілі, проводяться численні порівняння

декількох різних частот, кодованих масивом слухових рецепторів. Коли

відлуння отримується від об'єкта з невідомою затримкою, відповідає

той нейрон (слухової системи), який має відповідну затримку в лінії.

Таким чином оцінюється відстань до обсягу.

# ПОСТАНОВКА ЗАДАЧІ

Розробити та реалізувати метод, за допомогою якого розв’язується наступна задача:

Дані на вхід: - часовий ряд (послідовність значень деякого показника впорядкована по даті фіксування; передбачається, що фіксування значень виконується з однаковим інтервалом), де - елементи ряду,   
- кількість елементів ряду;

Вихід:- функція прогнозування, що будується відштовхуючись від даних на вхід; за аргументи має номер елементу ряду (може бути більшим і меншим за ), а за значення має t-ий елемент цього ряду.

Мають місце дані моніторингу первинної інвалідності в Україні за 24 роки (1992-2015). Для кожної адміністративної території, хвороби та типу населення дані представляють собою часовий ряд вигляду

,

де - значення первинної інвалідності на 10000 населення внаслідок хвороби , зафіксоване у t-му році на певний адміністративній території для однієї з верств населення; - кількість років, упродовж яких проводиться моніторинг (у даному випадку ).

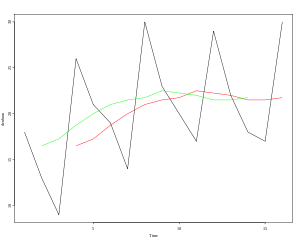
Ставиться задача прогнозування первинної інвалідності на наступний рік.

**Розділ 1. Прогнозування часових рядів**

**1.1.** Огляд методів прогнозування часових рядів

Метод ***ковзного середнього***( *moving average*, *MA*)*.*

Ковзне середнє - загальна назва для сімейства функцій, значення яких в кожній точці визначення дорівнюють середньому значенню початкової функції за попередній період. Ковзне середнє зазвичай використовуються з даними часових рядів для згладжування короткострокових коливань і виділення основних тенденцій або циклів. Математично ковзне середнє є одним з видів згортки.

 Рис.1. Просте ковзне середнє

Просте ковзне середнє, або арифметичне ковзне середнє ***SMA*** чисельно дорівнює середньому арифметичному значень вихідної функції за встановлений період і обчислюється за формулою:



де - значення простого ковзного середнього в точці ,- кількість значень початкової функції для розрахунку ковзного середнього, чим ширше згладжує інтервал, тим більш плавним виходить графік функції, значення вихідної функції в точці.

На Рис.1 зображені вихідна функція і її 2 прості ковзаючі середні по чотирьом значенням (n = 4): зелена лінія - центрування по середині інтервалу (справжній стан), червона лінія - зсув графіка вправо до останнього значенням вікна.

Експоненціально зважене ковзне середнє ***EMA*** - різновид зваженої ковзної середньої, ваги якої зменшуються експоненціально і ніколи не дорівнюють нулю. Визначається наступною формулою:



де - значення експоненціального ковзного середнього в точці , - значення експоненціального ковзного середнього в точці , - значення вихідної функції в момент часу , - коефіцієнт що характеризує швидкість зменшення вагів, приймає значення від 0 і до 1, чим менше його значення тим більше вплив попередніх значень на поточну величину середнього.

*(адаптивні)* ARIMA

“Гусениці”-SSA

локальна апроксимація (**LA**)

нечіткі

нейромережеві

Розділ 2. Програмне забезпечення прогнозування часових рядів

**2.1.** Функціональні можливості та структура програми

Програма складається з двох основних блоків: бібліотеки, що реалізує програмне забезпечення прогнозування часових рядів, користувацького інтерфейсу, що надає змогу скористатися можливостями бібліотеки без написання додаткових програм.

Бібліотека написана мовою **С++** і розділяється на дві логічні частини: незалежну від середовища, в якому буде виконуватися програма, і залежну. Таки спосіб розділення логіки надає змогу легко від’єднати незалежну частину бібліотеки і скористатися нею деінде.

Є такі незалежні від середовища виконування програмні блоки:

1. *matrix* задає і реалізує інтерфейс і алгебраїчні операції для матриці дійсних чисел;
2. Serialization описує і визначає функції для сериалізації і десериализації (кодування і розкодування у буфер) важливих для бібліотеки структур даних;
3. VectorAlgebra описує і визначає функції алгебри векторів;
4. Perceptron відповідає за реалізацію багатошарового перцептрону. Надає можливість навчити його будь-якою кількістю паттернів завдяки функції back\_prop. Також можна контролювали процес навчання завдяки функціям forward\_prop, put\_errors і flush, перша з яких відповідає за ForwardPropagation і запамятовує у буфері виходи кожного зі слоїв, друга за BackPropagation і наколює градієнт у буфері, а третя використовує цей градієнт для зміни вагів і очищує буфер. Використовувати ці функції при наявності масиву патернів треба так: (викликати forward\_prop, передати помилку у put\_errors) повторити для кожного патерну, викликати flush. Розділення способів навчання на більш простий і більш складний (1 і 3 функції) потрібне для того, щоб користувач об’єкта класу Perceptron мав змогу отримати помилки для першого слою перцептрону від функції put\_errors, тобто, якщо дані для навчання перцептрону не є правильними (початковими, незмінними), користувач, базуючись на отриманих помилках, має змогу змінити дані для навчання перцептрону. Для цієї можливості і був створений клас Perceptron, а не використана вже готова програма чи бібліотека, бо такої можливасті при навчанні нейронних мереж зазвичай не надають.  
   При закінченні навчання перцептрону користувачу об’єкта класу Perceptron рекомендується викликати функцію release\_buffer, що вивільняє усі ресурси, які були потрібні при навчанні перцептрону, але при використанні вже не є потрібними. Об’єкт класу Perceptron можна перевести у бінарний вигляд та зберігати у файлах чи бінарних змінних (сериалізувати); це надає змогу користувачеві відтворювати вже навчений перцептрон без витрачання часу на навчання. Cериализацию реалізують функції write\_to\_stream і from\_stream.
5. GammaUnit (гамма-юніт) прихована від користувача частина програми, але дуже важлива. Відповідає за згортку ряду, реалізує принцип гамма-пам’яті. Має змогу змінювати ваги гамма-пам’яті відповідно до величини помилки, що йому передалась. Цей логічний блок навчаеться у зв’язці з перцептроном: перцептрону передають данні на вхід, серед яких є значення згортки ряду від конкретногогамма-юніту; перцептрон повертає значення помилок вхідних даних; гамма-юніту змінює значення свого вагу або накопичує гражіент — залежить від кітькості патернів.
6. GammaNN обгортає Perceptron для використання з часовими рядами, надає і реалізує інтерфейс для навчання перцептрону та гамма-юнітів, можливість контролювати кількість гамма-юнітів (units) і кількість вільних входів (trace\_size), що бачать останні значення часового ряду. Має функцію learn для навчання, куди передаються номери елементів часового ряду, що є патернами для навчання. Має оператор [], для отримання об’єкту часового ряду жа його номером. Оптимізує використання [] таким чином, що запам’ятовує усі елементи ряду (елементи початкового ряду також), що були колись отримані. Ця оптимізація є вдалою, бо для визначення наступних значень часового ряду треба звертатися до попередніх, котрі треба було б теж обчислювати, якщо б їх не запам’ятали, і так доки не дійдемо до елементів початкового ряду. Як і Perceptron, GammaNN можна сериалізувати таким самим чином: функції write\_to\_stream і from\_stream.
7. Тестування підпрограм. Для тестування Percepton використовувалися навчальні дані задачі XOR та інші. Для тестування GammaNN використовувалися навчальні дані: послідовність числа 1, послідовність нулів та одиниць, двомірна послідовність нулів та одиниць, зростаюча послідовності чисел за кроком 1.

Залежність бібліотеки від середовища зумовлена тим, що було задумано створити бібліотеку для мови програмування **R** на якій був написаний інтерфейс.

Є такі функції, що експортуються у середовище мови **R**:

1. learn приймає на вхід початковій часовий ряд і параметри на для навчання GammaNN; віддає NNptr - вказівник на об’єкт класу GammaNN. З середовища мови **R** неможливо звертатися до членів об’єкту GammaNN, тому для нього цей вказівник виконує роль дескриптора, який потрібно передавати в експортованій бібліотекою інтерфейс користування.
2. get\_series приймає NNptr і номери обєектів часового ряду; віддає масив елементів часового ряді, номери яких прийшли на вхід.
3. get\_series\_length приймає NNptr; віддає кількість елементів ряду, що вже визначені в об’єкті GammaNN.
4. to\_GammaNN приймає строку (сериалізованний GammaNN об’єкт); віддає NNptr.
5. to\_str обернено до to\_GammaNN
6. create\_from\_file приймає путь до файлу, де сериалізований GammaNN об’єкт; віддає NNptr.
7. write\_to\_file приймає NNptr, путь до файлу, куди сериалізовати GammaNN об’єкт.

інтерфейс користувача створений на мові **R** з використанням програмного пакету **Shiny**. Таке поєднання визначає структуру частини програми відповідальної за інтерфейс. Є два логічних блоки, що відповідають за інтерфейс користувача: UI і server. Інтерфейс користувачя в даному випадку — клієнт-серверна програма. UI відповідає за розміщення елементів інтерфейсу на екрані, а server за обробку даних.

# ТЕСТУВАННЯ ПРОГРАМНОГО ПРОДУКТУ

# ВИСНОВКИ

# СПИСОК ЛІТЕРАТУРИ

* <http://chaos.phys.msu.ru/loskutov/PDF/Lectures_time_series_analysis.pdf>
* Хайкин. “Нейронные сети полный курс”, 2-е издание. Издательский дом “Вильямс”, 2006. - 1104с.
* [https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D0%BE%D0%BB%D1%8C%D0%B7%D1%8F%D1%89%D0%B0%D1%8F\_%D1%81%D1%80%D0%B5%D0%B4%D0%BD%D1%8F%D1%8F#cite\_note-itl.nist.gov.pmc324-6](https://ru.wikipedia.org/wiki/Скользящая_средняя" \l "cite_note-itl.nist.gov.pmc324-6)