ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ

дисциплины Анализ и прогнозирование временных рядов методами искусственного интеллекта
для направления 09.04.04 Программная инженерия
уровень образования магистратура
профиль подготовки Искусственный интеллект и инженерия данных
форма обучения очная
кафедра-разработчик Системное программирование

Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 09.04.04 Программная инженерия, утверждённым приказом Минобрнауки от 19.09.2017 № 932

Разработчик программы,
доктор физ-мат. наук, доцент _____ М.Л. Цымблер
(ученая степень, ученое звание, должность) (подпись)

Зав. кафедрой Системное программирование
д.физ-мат.н., проф. _____ Л. Б. Соколинский
(ученая степень, ученое звание) (подпись)

Челябинск-2021

Оглавление

1. Выполнение работ

Задание, выполняемое на практическом занятии, предполагает решение студентом небольшой учебно-исследовательской задачи по теме дисциплины, подготовку и защиту отчета о разработанном решении. Задача, как правило, заключается в выполнении интеллектуального анализа указанного набора данных (временного ряда) и визуализации полученных результатов.

Студенту необходимо создать на одном из свободно доступных сервисов (github, bitbucket и др.) публичный *репозиторий по дисциплине* для сохранения исходных текстов разработанных решения заданий и др. материалов, создаваемых в рамках практических занятий.

Набор данных предлагается студентом и согласовывается с преподавателем, при этом предпочтительны референсные наборы данных – размещенные в авторитетных свободно доступных интернет-репозиториях (например, UCR Time Series Classification Archive https://www.cs.ucr.edu/~eamonn/time_series_data_2018/ или UCI Machine Learning Repository https://archive.ics.uci.edu/) или/и упомянутые в научных статьях, опубликованных в авторитетных рецензируемых журналах.

Алгоритм интеллектуального анализа данных может быть реализован студентом с помощью сторонних библиотек или самостоятельно (предпочтительно). При разработке *программы* допустимо использовать любые языки программирования, библиотеки и инструментальные средства (если явно не указано обратное).

Исходные тексты программы и сопутствующие материалы задания (наборы данных, результаты работы и визуализации, отчет) необходимо сохранять в репозитории по дисциплине (отдельный каталог для каждого задания). Исходные тексты должны быть документированы (наличие спецификаций файлов и подпрограмм).

Дополнительные (бонусные) баллы: за качественное использование в реализации параллельных/распределенных алгоритмов.

Отчет о выполнении задания должен включать в себя следующие основные элементы:

- полные ФИО автора отчета, адрес электронной почты для связи;
- формулировка задания;
- библиографическая ссылка и краткие сведения о наборе данных;
- краткие сведения о средствах реализации (если применимо) и гиперссылка на каталог репозитория с исходными текстами и сопутствующими материалами;

– рисунки с результатами визуализации[1];
– краткие пояснения к полученным результатам.

Защита отчета предполагает устные ответы студента на вопросы преподавателя по реализации программы и полученным результатам.

[1] Рисунки должны иметь подписи. Графики и диаграммы на рисунках должны иметь легенду, подписи осей с указанием единиц измерения (если применимо).

2. Поиск подпоследовательностей по образцу

Задание 1. Поиск с использованием меры DTW

1. Разработайте программу, которая выполняет поиск top-k подпоследовательностей временного ряда, похожих на образец поиска в смысле меры DTW (Dynamic Time Warping), без использования оптимизаций (раннее отбрасывание, каскадное применение нижних границ и др.). Параметрами программы являются ряд, образец поиска, число k.
2. Проведите эксперименты на трех временных рядах из различных предметных областей и пяти образцов поиска различной длины (взяв $k = 5$).
3. Выполните визуализацию результатов экспериментов в следующем виде:

- наложение на одной диаграмме найденных подпоследовательностей и образца поиска (показанных различными цветами) с указанием значений схожести результата с образцом в смысле меры DTW и евклидова расстояния;
- диаграмма сравнения быстродействия поиска на фиксированном ряде при изменяемой длине образца.

4. Проанализируйте и изложите содержательный смысл полученных результатов.

Задание 2. Поиск с использованием подхода UCR-DTW

1. Разработайте программу, которая выполняет поиск top-k подпоследовательностей временного ряда, похожих на образец поиска в смысле меры DTW, с использованием оптимизаций UCR-DTW (раннее отбрасывание, каскадное применение нижних границ и др.). Параметрами программы являются ряд, образец поиска, число k, число r – ширина полосы Сако–Чиба (как доля от длины образца).
2. Проведите эксперименты на ранее выбранных временных рядах и образцах поиска для $r = \{0.1, 0.5, 0.8, 1.0\}$ (взяв $k = 5$).
3. Выполните визуализацию результатов экспериментов в следующем виде:

- наложение на одной диаграмме найденных подпоследовательностей и образца поиска (показанных различными цветами) с указанием значений схожести результата с образцом в смысле меры DTW и евклидова расстояния (отдельно для каждого значения параметра r) ;
- диаграмма сравнения быстродействия поиска на фиксированном ряде при изменяемой длине образца (отдельно для каждого значения параметра r).

4. Проанализируйте и изложите содержательный смысл полученных результатов, в т.ч. сравнив подходы DTW (по результатам задания 1) и UCR-DTW.

3. Поиск аномалий во временных рядах

Задание 3. Поиск диссонансов с помощью алгоритма DRAG

1. Разработайте программу, которая выполняет поиск top-k диссонансов временного ряда, используя алгоритм DRAG. Параметрами программы являются ряд, длина диссонанса, число k, число r – минимальное расстояние диссонанса до его ближайшего соседа.
2. Проведите эксперименты на трех временных рядах из различных предметных областей, пяти различных длин диссонанса и пяти различных значений параметра r (взяв $k = 5$).
3. Выполните визуализацию результатов экспериментов в следующем виде:

- отображение временного ряда, в котором диссонансы показаны различными цветами (отличными от цвета ряда; отдельно для каждого значения параметра r);
- наложение на одной диаграмме пар «найденный диссонанс, ближайший сосед диссонанса» (показанных различными цветами) с указанием значений схожести диссонанса и его ближайшего соседа в смысле евклидова расстояния (отдельно для каждого значения параметра r);
- диаграмма сравнения быстродействия на фиксированном ряде при изменяемой длине диссонанса (отдельно для каждого значения параметра r).

4. Проанализируйте и изложите содержательный смысл полученных результатов.

Задание 4. Поиск диссонансов с помощью алгоритма MERLIN

1. Разработайте программу, которая выполняет поиск top-k диссонансов временного ряда, используя алгоритм MERLIN. Параметрами программы являются ряд, диапазон длин диссонанса и число k.
2. Проведите эксперименты на ранее выбранных временных рядах, взяв диапазон длин диссонанса таким, чтобы он покрывал длины диссонансов, задействованные в экспериментах с алгоритмом DRAG задания 3 (взяв $k = 5$).

- отображение временного ряда, в котором диссонансы показаны различными цветами (отличными от цвета ряда);
- наложение на одной диаграмме пар «найденный диссонанс, ближайший сосед диссонанса» (показанных различными цветами) с указанием значений схожести диссонанса и его ближайшего соседа в смысле евклидова расстояния (отдельно для каждого значения длины диссонанса, задействованного в экспериментах с алгоритмом DRAG);

- таблица, показывающая индексы соответствующих (по рангу) диссонансов, найденных с помощью алгоритмов DRAG и MERLIN.
- диаграмма сравнения быстродействия поиска на фиксированном ряде при изменяемой длине диссонанса.

3. Выполните визуализацию результатов экспериментов в следующем виде:
4. Проанализируйте и изложите содержательный смысл полученных результатов, в т.ч. сравнив алгоритмы DRAG (по результатам задания 3) и MERLIN.

4. Матричный профиль временного ряда и примитивы анализа данных на его основе

Задание 5. Вычисление матричного профиля

1. Разработайте программу, которая выполняет вычисление матричного профиля временного ряда с помощью алгоритма SCAMP. Параметрами программы являются ряд и длина подпоследовательности.
2. Проведите эксперименты на ранее выбранных временных рядах и пяти различных длин подпоследовательности.
3. Выполните визуализацию результатов экспериментов в следующем виде:

- одновременное отображение временного ряда и его матричного профиля;
- диаграмма сравнения быстродействия вычислений на фиксированном ряде при изменяемой длине подпоследовательности.

4. Проанализируйте и изложите содержательный смысл полученных результатов.

Задание 6. Поиск диссонансов с помощью матричного профиля ряда

1. Доработайте программу вычисления матричного профиля временного ряда (см. задание 5) таким образом, чтобы она выполняла также поиск top-k диссонансов по вычисленному матричному профилю ряда. Параметрами программы являются ряд, длина подпоследовательности (диссонанса) и число k.
2. Проведите эксперименты, используя данные и параметры из задания 3.
3. Выполните визуализацию результатов экспериментов в следующем виде:

- отображение временного ряда, в котором диссонансы показаны различными цветами (отличными от цвета ряда);
- таблица, показывающая индексы соответствующих (по рангу) диссонансов, найденных с помощью матричного профиля, алгоритмов DRAG (см. задание 3) и MERLIN (см. задание 4).

4. Проанализируйте и изложите содержательный смысл полученных результатов.

Задание 7. Поиск мотивов с помощью матричного профиля ряда

1. Доработайте программу вычисления матричного профиля временного ряда (см. задание 5) таким образом, чтобы она выполняла также поиск top-k мотивов по вычисленному матричному профилю ряда. Параметрами программы являются ряд, длина подпоследовательности (мотива) и число k.
2. Проведите эксперименты по поиску мотивов, используя данные и параметры из задания 5.

3. Выполните визуализацию результатов экспериментов в следующем виде:
- одновременное отображение матричного профиля и соответствующего временного ряда, в котором мотивы (левые и правые части) показаны различными цветами (отличными от цвета ряда);
- диаграмма сравнения быстродействия вычислений на фиксированном ряде при изменяемой длине мотива.
4. Проанализируйте и изложите содержательный смысл полученных результатов.

Задание 8. Поиск типичных подпоследовательностей с помощью матричного профиля ряда

1. Разработайте программу, которая выполняет поиск top-k типичных подпоследовательностей временного ряда (сниппетов) с помощью алгоритма SnippetFinder. Параметрами программы являются ряд, длина подпоследовательности (сниппета) и число k.
2. Проведите эксперименты на трех временных рядах из различных предметных областей и пяти различных длин сниппета (взяв $k = 5$).
3. Выполните визуализацию результатов экспериментов в следующем виде:
- отображение временного ряда, в котором сниппеты показаны различными цветами, и в легенде указана значимость каждого сниппета.
4. Проанализируйте и изложите содержательный смысл полученных результатов.

Задание 9. Поиск эволюционирующих шаблонов с помощью матричного профиля ряда

1. Разработайте программу, которая выполняет поиск эволюционирующих шаблонов временного ряда (цепочек) с помощью алгоритма ALLC (All Chain). Параметрами программы являются ряд и длина подпоследовательности (звена цепочки).
2. Проведите эксперименты на трех временных рядах из различных предметных областей и пяти различных длин подпоследовательности (взяв $k = 5$).
3. Выполните визуализацию результатов экспериментов в следующем виде:
- отображение временного ряда, в котором звенья цепочки показаны различными цветами (отличными от цвета ряда);
- отображение звеньев цепочки на одной диаграмме с указанием индексов начала соответствующих подпоследовательностей.
4. Проанализируйте и изложите содержательный смысл полученных результатов.

5. Восстановление пропусков и прогноз значений временного ряда

Задание 10. Восстановление пропусков с помощью аналитических алгоритмов

1. Разработайте программу, которая выполняет восстановление пропущенных значений временного ряда с помощью аналитического алгоритма и оценивает точность восстановления. Параметрами программы являются ряд, размер окна восстанавливаемых точек в конце ряда, используемые меры оценки точности, а также параметры конкретного алгоритма восстановления. Для реализации используйте любые пять алгоритмов из следующего списка: HotDeck, Mean Imputation, Mode imputation, TKCM, REBOM, MUSCLES, SPIRIT, DynaMMo. Подберите три любых меры оценки точности восстановления из следующего списка: MSE (Mean Squared Error), RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error), APE (Absolute Percentage Error), R2 (коэффициент детерминации).
2. Проведите эксперименты на трех временных рядах из различных предметных областей.
3. Выполните визуализацию результатов экспериментов в следующем виде:

- отображение временного ряда, в котором реальные и восстановленные значения показаны различными цветами;
- таблица, показывающая точность восстановления выбранными алгоритмами в смысле выбранных мер.

4. Проанализируйте и изложите содержательный смысл полученных результатов.

Задание 11. Прогнозирование временного ряда с помощью модели ARIMA

1. Разработайте программу, которая выполняет прогнозирование значений временного ряда с помощью модели ARIMA (Autoregressive Integrated Moving Average) или ее модификаций (SARIMAX и др.) и оценивает точность прогноза.
2. Опишите процесс подбора параметров модели.
3. Проведите эксперименты, используя данные из задания 10.
4. Выполните визуализацию результатов экспериментов в следующем виде:

- отображение временного ряда, в котором реальные и прогнозные значения показаны различными цветами;

- таблица, показывающая точность прогноза с помощью ARIMA в сравнении с аналитическими алгоритмами в смысле выбранных мер (см. задание 10).

5. Проанализируйте и изложите содержательный смысл полученных результатов.

Задание 12. Прогнозирование временного ряда с помощью рекуррентной нейронной сети

1. Разработайте программу, которая выполняет прогнозирование значений временного ряда с помощью рекуррентной нейронной сети, целиком состоящей из длинной цепи элементов краткосрочной памяти (Long Short-Term Memory, LSTM) или управляемых рекуррентных блоков (Gated Recurrent Units, GRU) и оценивает точность прогноза.
2. Опишите процесс подбора параметров и обучения нейронной сети, приведите рисунок с топологией разработанной нейронной сети.
3. Проведите эксперименты, используя данные из задания 10.
4. Выполните визуализацию результатов экспериментов в следующем виде:

- отображение временного ряда, в котором реальные и восстановленные значения показаны различными цветами;
- таблица, показывающая точность восстановления с помощью построенной нейронной сети в сравнении с моделью ARIMA (см. задание 11) и аналитическими алгоритмами в смысле выбранных мер (см. задание 10).

5. Проанализируйте и изложите содержательный смысл полученных результатов, в т.ч. укажите характеристики и предметную область временного ряда, влияющие на выбор подхода, используемого для восстановления/прогноза значений ряда.

Основная литература

1. Aggarwal C.C. Data Mining: The Textbook. Springer, 2015. 746 p. ISBN 978-3-319-14141-1. Chapter 14. Mining Time Series Data, P. 457-493. https://doi.org/10.1007/978-3-319-14142-8
2. Cryer J.D., Chan K.-S. Time Series Analysis with Applications in R. 2nd Edition. 506 p. ISBN: 978-0-387-75958-6

Дополнительная литература

1. Imani S., Madrid F., Ding W., Crouter S.E., Keogh E.J. Introducing time series snippets: a new primitive for summarizing long time series // Data Min. Knowl. Discov. 2020. Vol. 34, no. 6. P. 1713-1743. https://doi.org/10.1007/s10618-020-00702-y
2. Nakamura T., Imamura M., Mercer R., Keogh E.J. MERLIN: Parameter-Free Discovery of Arbitrary Length Anomalies in Massive Time Series Archives // Proceedings of the 20th IEEE International Conference on Data Mining, ICDM 2020, Sorrento, Italy, November 17-20, 2020. IEEE, 2020. P. 1190-1195. https://doi.org/10.1109/ICDM50108.2020.00147
3. Rakthanmanon T., Campana B.J.L., Mueen A., Batista G.E.A.P.A., Westover M.B., Zhu Q., Zakaria J., Keogh E.J. Addressing Big Data Time Series: Mining Trillions of Time Series Subsequences Under Dynamic Time Warping // ACM Trans. Knowl. Discov. Data. 2013. Vol. 7, no. 3. P. 10:1-10:31. https://doi.org/10.1145/2500489
4. Zhu Y., Gharghabi S., Silva D.F., Dau H.A., Yeh C.-C.M., Senobari N.S., Almaslukh A., Kamgar K., Zimmerman Z., Funning G.J., Mueen A., Keogh E.J. The Swiss army knife of time series data mining: ten useful things you can do with the matrix profile and ten lines of code // Data Min. Knowl. Discov. 2020. Vol. 34, no. 4. P. 949-979. https://doi.org/10.1007/s10618-019-00668-6
5. Yankov D., Keogh E.J., Rebbapragada U. Disk aware discord discovery: finding unusual time series in terabyte sized datasets // Knowl. Inf. Syst. 2008. Vol. 17, no. 2. P. 241-262. https://doi.org/10.1007/s10115-008-0131-9
6. Yeh C.-C.M., Zhu Y., Ulanova L., Begum N., Dau H.A., Silva D.F., Mueen A., Keogh E.J. Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets // Proceedings of the IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain. IEEE, 2016. P. 1317-1322. https://doi.org/10.1109/ICDM.2016.0179
7. Zhu Y., Imamura M., Nikovski D., Keogh E.J. Matrix Profile VII: Time Series Chains: A New Primitive for Time Series Data Mining // Proceedings of the 2017 IEEE International Conference on Data Mining, ICDM 2017, New Orleans, LA, USA, November 18-21, 2017. IEEE, 2017. P. 695-704. https://doi.org/10.1109/ICDM.2017.79