Дизайн систем машинного обучения

12. Временные ряды и графы

План курса

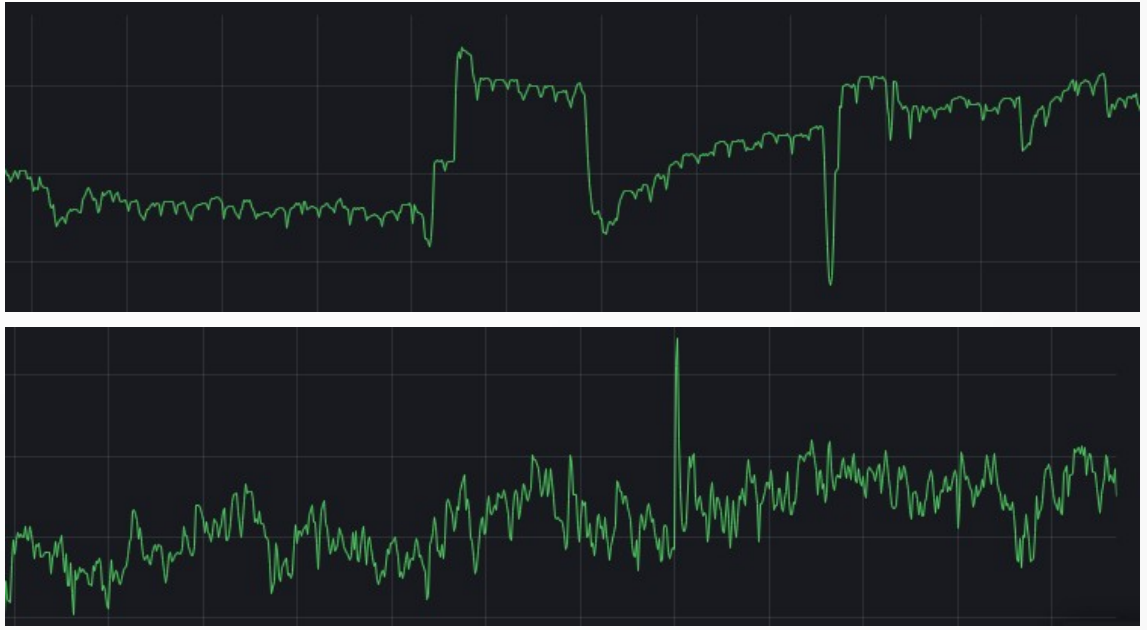
1) ~~Практическое применение машинного обучения~~
2) ~~Основы проектирования ML-систем~~
3) ~~Обучающие данные~~
4) ~~Подготовка и отбор признаков~~
5) ~~Выбор модели, разработка и обучение модели~~
6) ~~Оценка качества модели~~
7) ~~Развертывание~~
8) ~~Диагностика ошибок и отказов ML-систем~~
9) ~~Мониторинг и обучение на потоковых данных~~
10) ~~Жизненный цикл модели~~
11) ~~Отслеживание экспериментов и версионирование моделей~~
12) **Сложные модели: временные ряды, модели над графами — Вы находитесь здесь**
13) Непредвзятость, безопасность, карточки моделей
14) ML инфраструктура и платформы
15) Интеграция ML-систем в бизнес-процессы

Что мы знаем про данные

- Как устроены
- Откуда берутся
- Как собираем
- Где храним
- Детекция аномалий — как искать
- Пропущеные значения — как заполнить
- Сэмплирование — как мы можем взять репрезентативную выборку
- Предсказания — что обычно предсказывают
- Эмбеддинги — как построить векторное представление

Временные ряды Time Series

- Как устроены
- Откуда берутся
- Как собираем
- Где храним
- Детекция аномалий
- Пропущеные значения
- Сэмплирование
- Предсказания
- Эмбеддинги



Временные ряды

- Последовательность однотипных значений,
  индексированных временем →
- Обычно — количественные.
  Могут быть порядковыми и номинальными.
- Обычно одно значение, может быть несколько
- Могут быть взяты через равные промежутки времени
  или через произвольные промежутки
- Разные переменные могут быть взяты с разной частотой
- Время может быть указано с разной точностью и в разных часовых поясах

Временные ряды - примеры

- Измеряемые величины (имеют мгновенное значение)
  - Скорость вращения двигателя
  - Энергопотребление
  - Температура сервера
  - Курс акций
  - Ранг сайта в поисковой выдаче (порядковый)
  - Состояние замка (отрыт-закрыт, номинальный)
- Аггрегированные значения (не имеют смысла без указания временного отрезка)
  - Сумма продажи
  - Количество посетителей на сайте

Время в данных

- Часовые пояса могут быть указаны или не указаны
- Часовые пояса влияют на дату
- Желательно иметь единый часовой пояс в системе (UTC +0, будут проблемы)
- Если все сервера в одной локации — можно использовать время этой локации (будут проблемы)
- Если источник данных не сообщает часовой пояс, нужно добавить его как в данные при сохранении (будут проблемы все равно)
- Часовой пояс источника данных может меняться (DST и т. д.)
- Библиотеки часто конвертируют часовые пояса внутри себя, по-разному от версии к версии (привет, PyMongo)
- См Заблуждения программистов о времени

Временные ряды — как собираем

- Если источников мало — могут писать в базу данных сами
- Чаще всего данные приходят в API
- Типа «дата» нет в JSON (но есть в BSON) - приходит строкой
- Удобно складывать временные ряды в топики Kafka
  - Использовать их прямо из Kafka, не из базы
  - Считать оконные статистики перед записью в базу
  - Пересемплировать данные перед записью в базу
  - Накапливать данные перед записью в базу

Где храним

- RDBMS (PostgreSQL, Oracle, MS SQL etc)
- Колоночные базы данных (Clickhouse)
- Файловая система (S3)
- AWS Timestream
- Non-SQL базы (MongoDB, Redis)
- TSDB Time Series Databases (Influx, TimescaleDB)
- Hadoop (OpenTSDB/HBase)
- BigQuery (дешевле, мощнее) / BigTable (быстрее, дороже)
- YDB Yandex DataBase/ AWS Dynamo DB

Данных много

- Делим данные на части для хранения (партиции)
- Естественное партиционирование — по времени
- Проблема — все время пишем в последнюю партицию

Что выбрать?

- То, что уже есть в системе
- Если данных мало — можно в RDBMS
- Если есть Hadoop - OpenTSDB
- Если данные сложные — MongoDB (?)
- Если часто пишем и редко читаем - ClickHouse
- Если данных очень много — BigQuery / Timestream

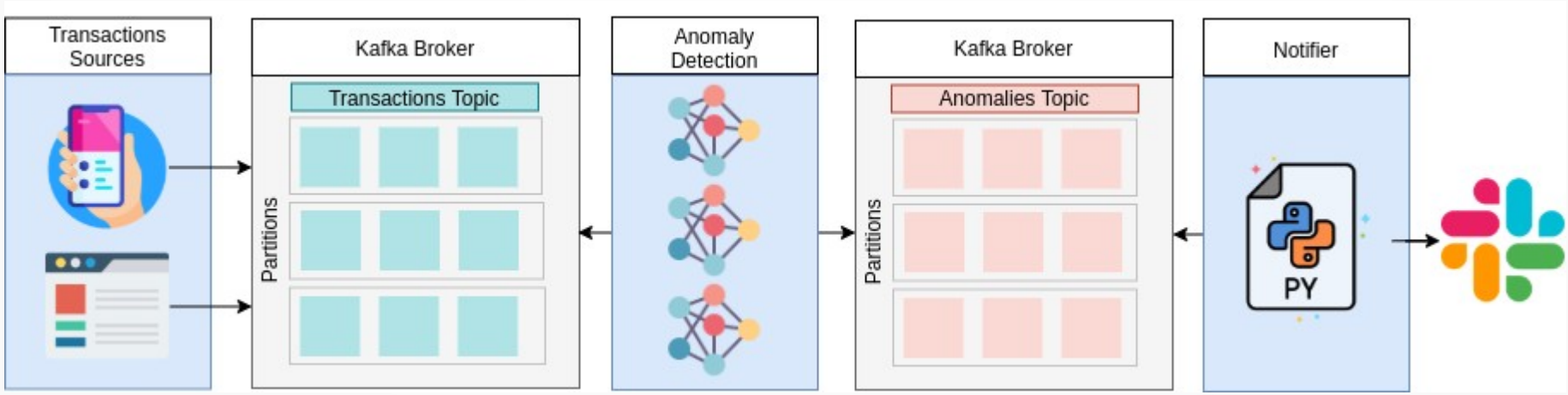
Как заливать данные

- TimescaleDB: Kafka → TimescaleDB
- ClickHouse : Kafka → ClickHouse
- Redis: Kafke → Redis
- MongoDB: Kafka → MongoDB
- Timestream: Kinesis → Timestream
- BigQuery: Dataflow (Apache Beam) → BigQuery

Аномалии с Kafka

С Kafka вручную любой моделью, см

Real-time anomaly detection with Apache Kafka and Python



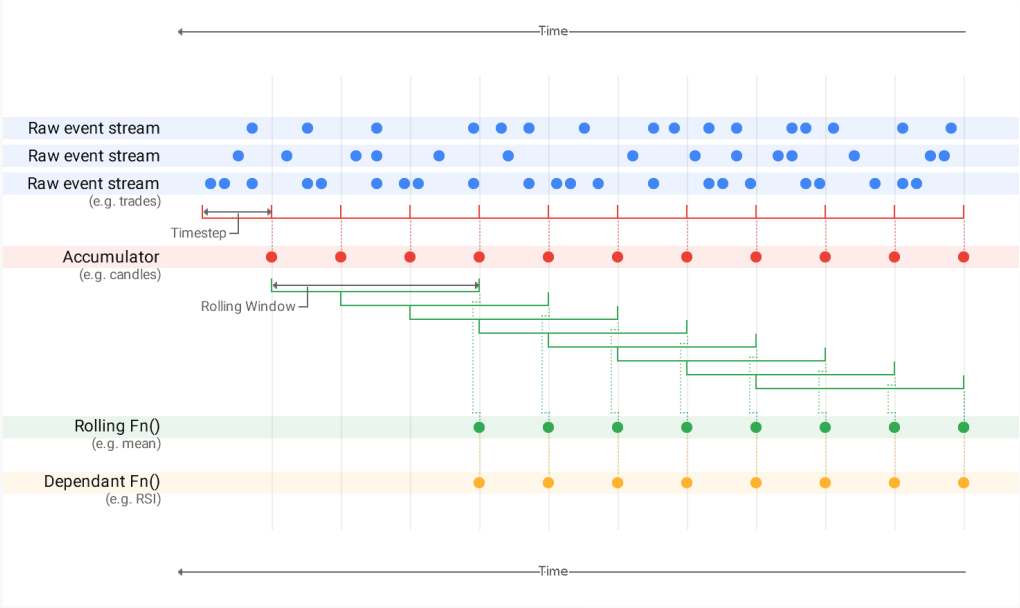
Предсказания с Apache Beam

- Parallel forecasting with Apache beam and Prophet

- Pub/Sub → DataFlow → BigQuery
- BigQuery → DataFlow → Prophet → BigQuery

Примеры: торговый робот, IoT

- Dataflow-sample-applications: Timeseries-streaming



Анализ временных рядов

- Много старой доброй статистики (SARIMA и т. д.)
- Деревья — часто используют
- Etna - time series forecasting framework
- Prophet - Forecasting at scale, Stan
- Luminaire - Monitoring Time Series Data
- GluonTS - Probabilistic Time Series Modeling in Python
- PyTorch Forecasting with Temporal Fusion Transformer

Бейзлайн для временных рядов

- Завтра будет так же, как вчера

$$T_{n+1} = T_n$$

- Завтра будет так же, как год назад

$$T_{n+1} = T_{n+1-365}$$

- Завтра будет такой же прирост

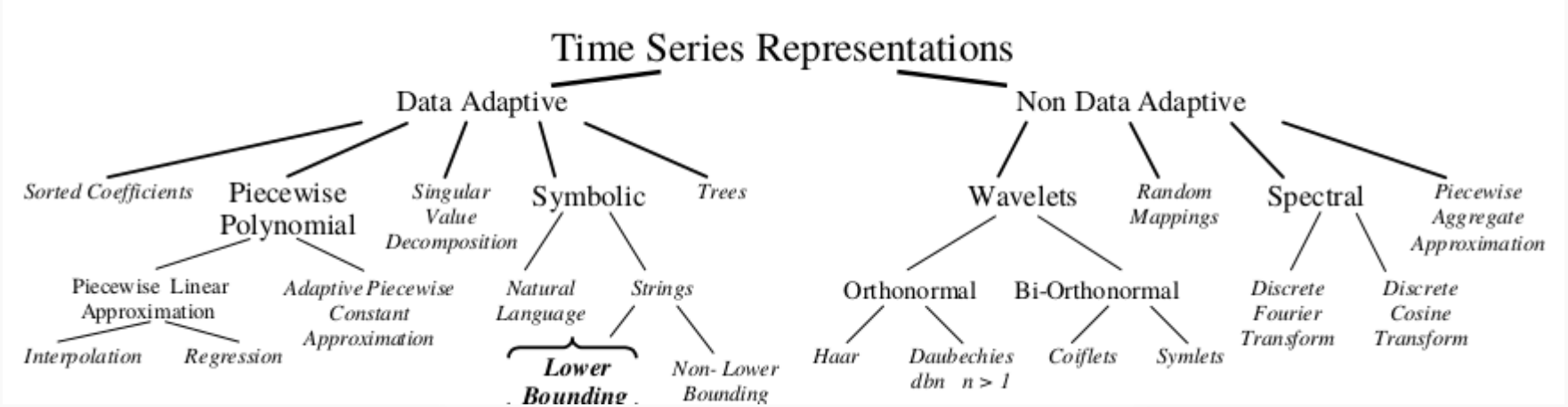
$$T_{n+1} = 2T_n - T_{n-1}$$

Сэмплинг

- Интерполировать или брать ближайший
  - Если точки с переменным шагом:
  - Если значения пропущены
  - Если нужно с другой частотой
- A Survey on Principles, Models and Methods for Learning from Irregularly Sampled Time Series →
- https://en.wikipedia.org/wiki/Upsampling
- https://en.wikipedia.org/wiki/Downsampling_(signal_processing)
-

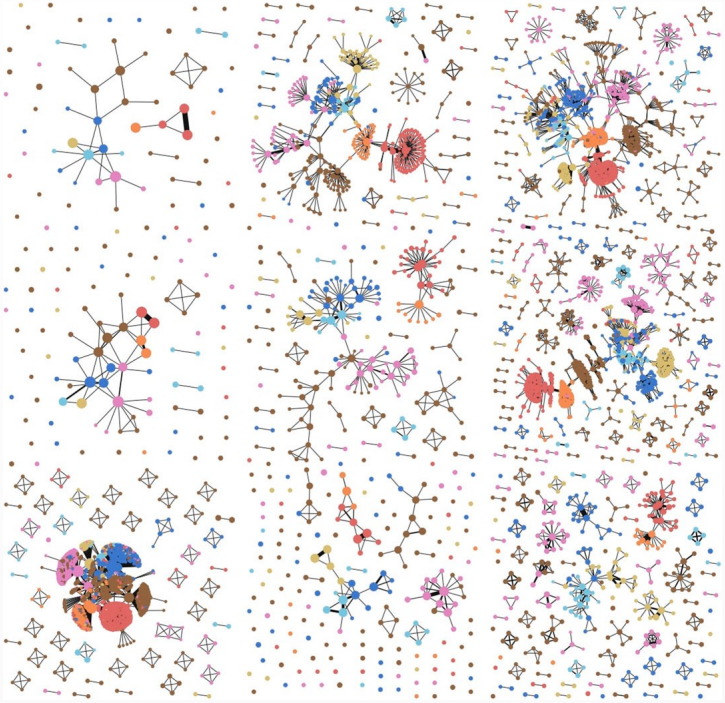
Временные ряды как тексты

- A Symbolic Representation of Time Series,
  with Implications for Streaming Algorithms, в доп. материалах



Графовые данные

- Как устроены
- Откуда берутся
- Как собираем
- Где храним
- Детекция аномалий
- Пропущеные значения
- Сэмплирование
- Предсказания
- Эмбеддинги



Граф из статьи Deep Graph Mapper: Seeing Graphs Through the Neural Lens

Графовые данные

- Данные, описывающие отношения между сущностями
- Граф состоит из вершин и ребер
  - Ориентированный / неориентированный
  - Циклический / ациклический
  - С аттрибутами в узлах и ребрах, в т.ч. взвешенный
  - И еще много интересной математики и терминологии

Примеры графов

- Цитирование в научных статьях
- Ссылки на веб-страницах
- Коммуникация в социальных сетях
- Дорожный граф на карте
- Маршруты транспорта
- Графы бенефициаров организаций
- Графы денежных потоков
- Графы взаимосвязей систем

Где возникают графовые данные

- Везде, где у нас есть две и более сущности в одной записи
- Транзакционные данные (финансы)
- Данные о структуре (биология, химия)
- Журналы работы (обычно process mining)
- Многое можно с пользой представить как граф
- Графовый анализ — обзор и области применения

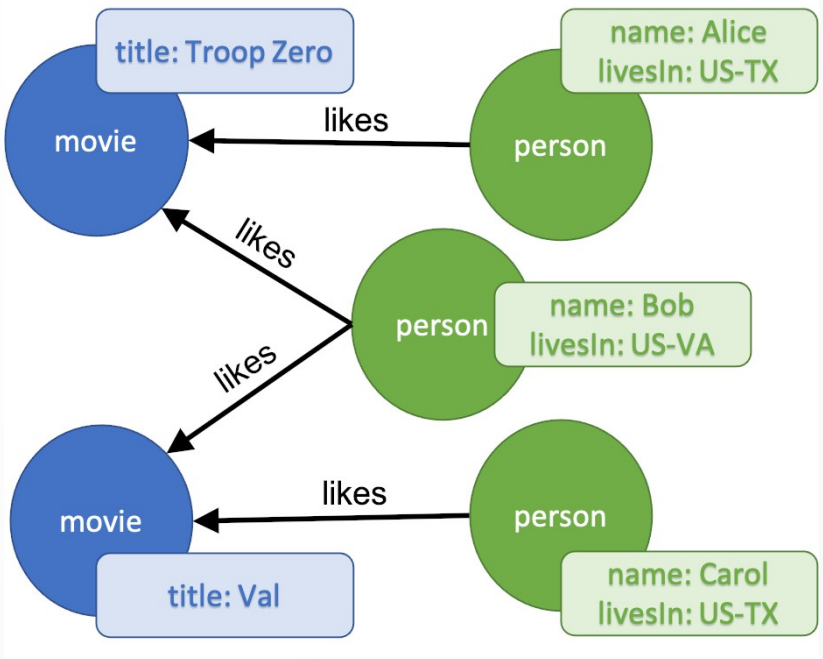
Как собирать графовые данные

- Обычно графовые данные приходят как табличные данные
- Clickstream — поток событий о переходах пользователя
- Поток транзакций
- Все как обычно — сначала складываем в kafka

Как работать с графами

- RDBMS (Рекурсивные запросы)
- MongoDB (GraphLookup)
- Neo4j
- Apache TinkerPop

```
g.V().hasLabel('movie')
.has('title','Troop Zero')
.in('likes')
.has('livesIn','US-TX')
.count()
```



https://aws.amazon.com/blogs/opensource/why-developers-like-apache-tinkerpop-an-open-source-framework-for-graph-computing/

Нейронные сети

- Статьи:
  - DeepWalk →
  - LINE →
  - HARP →
  - Graph sampling for node embedding →
- Graph Embedding в PapersWithCode →
- Инструменты
  - Deep Graph Library →
  - PyTorch Geometric →

Что можно делать с графами

- Использовать графовые эмбеддинги как признаки в моделях
- Предсказывать новые ребра графа
  - Предлагать друзей в соцсети
  - Подгружать в фоновом режиме страницу для просмотра
- Предсказывать изчезновение ребер
  - Отказ от использования сервиса
- Искать кратчайшие пути
- Искать устойчивые подграфы
- Process Minning

Дополнительные материалы

- Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting →
- TS2Vec: Towards Universal Representation of Time Series →
- A Symbolic Representation of Time Series, with Implications for Streaming Algorithms →

Все будет в телеграм-канале