*(слайд 1)*

Добрий день, мене звати Антон Мишенін. Сьогодні я представляю свою дипломну роботу на ступінь бакалавра на тему «Реалізація веб-застосунку для візуалізації багатовимірних даних з застосуванням алгоритму t-SNE».

*(слайд 2)*

Візуалізація була і є одним з найефективніших методів дослідження даних. Щоб багато не говорити, продемонструю істинність цього твердження на практиці. З лівого богу, ви бачити таблицю цін акцій компанії Motorola Solution Inc. в період з 13 березня по 13 квітня 2018 року. Без сумніву, ця таблиця містить в собі всі необхідні дані, але чи є вона інформативна? Що можна сказати про становище компанії просто дивлячись на цю таблицю? Майже нічого. На противагу їй, на правій стороні слайду ви бачите ті ж самі данні, але представлені у вигляді графіку. Незважаючи на те, що ціни ніяким чином на ньому не представлені, ми можемо стверджувати, що десь до 20ого березня, в компанії був період відносної стабільності, з 20 березня до 6 квітня – період загального спаду з тимчасовими підйомами, що не перевищують значень в період до 20 березня, з 6 квітня ми бачимо ріст і початок періоду стабільності, однак, ціна не змогла повернутись на до кризисні позначки.

Однак не для всіх наборів даних візуалізація настільки тривіальна. В цьому випадку у нас було лише дві змінні – дата та ціна, з яких з легкістю можна побудувати графік. Але що робити тоді, коли природа описуваних об’єктів набагато складніша, і нам необхідно зобразити більше двох, і навіть, більше трьох змінних? На це питання і відповідає алгоритм t-SNE, що розшифровують як t-distributed stochastic neighbor embedding.

*(слайд 3)*

t-SNE – це алгоритм машинного навчання, призначений для зниження розмірності багатовимірних даних з метою їх подальшої візуалізації. Він складається з двох етапів, етап І ­– побудувати розподіл ймовірностей для кожної пари багатовимірних об’єктів, таким чином, що схожі об’єкти мають високу вірогідність бути згрупованими, а несхожі – малу. На другому етапі, будують подібний розподіл ймовірностей на відповідній низько вимірній мапі, а між самими розподілами мінімізується Відстань Кульбака — Лейблера, враховуючи місцезнаходження точок.

*(слайд 4)*

Зважаючи на складність алгоритму, найкращій спосіб його пояснити – це пояснити на прикладі. Припустимо, маємо двовимірний простір, тобто площину, на якій зображені точки. Кожна точка являє собою сукупність двох величин, так само як в прикладі з цінами на акції Motorola Solution Inc. Наша задача – перевести двовимірні точки в одновимірні, тобто точку на прямій.

Як я вже говорив раніше, на першому етапі треба побудувати розподіл ймовірностей для кожної пари багатовимірних об’єктів, для цього:

1. визначимо точку, в якій ми зацікавлені\*
2. визначимо відстань між точкою, якою ми зацікавлені, і всіма іншими\*
3. підставити відстань в якості аргументу функції нормального розподілу Гауса, отримані значення прийняти за непромаштабований коефіцієнт схожості\*
4. промаштабувати ці коефіцієнти, як відношення непромаштабованого коефіцієнта схожості для пари точок і суми всіх коефіцієнтів схожості для точки зацікавлення.

*(слайд 5)*

Ці дії повторяти доти, доки всі точки не побувають у ролі точки зацікавлення. З отриманих чисел будуємо матрицю схожості, схожість точки самої на себе приймаємо як 0. На цьому ми завершуємо перший етап.

*(слайд 6)*

На другому етапі, будуємо карту на низькому вимірі з випадково розставленими точками. Так само як і минулого разу формуємо матрицю схожостей, але використовуємо не нормальний розподіл Гауса, а т-розподіл Стюарда.

*(слайд 7)*

Далі ми пересуваємо точки на прямій і відповідні комірки в другій матриці таким чином, зоб перша і друга матриці були тотожні. Звичайно, це дещо спрощена інтерпретація алгоритму, але вона максимально розкриває суть того, що відбувається.

*(слайд 8)*

В якості імплементації алгоритму t-SNE, я використав бібліотеку tsne-js авторства користувача сервісу github scienceai.