

# Высокопроизводительные компьютерные технологии

Задача 1. Gzip classification

# Суть алгоритма

- Между элементами тренировочного и тестового датасетов считается нормализованная дистанция сжатия **NCD**

$$NCD(x, y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

Здесь  $x, y$  – некоторые последовательности,  $xy$  – конкатенация,  
 $C(\cdot)$  – это длина последовательности сжатой с **gzip**

- Затем применяется метод k-ближайших соседей для определения классов тестового датасета
- Метод работает на любых типах данных

# Код из оригинальной статьи

```
1 import gzip
2 import numpy as np
3 for (x1, _) in test_set:
4     Cx1 = len(gzip.compress(x1.encode()))
5     distance_from_x1 = []
6     for (x2, _) in training_set:
7         Cx2 = len(gzip.compress(x2.encode()))
8         x1x2 = " ".join([x1, x2])
9         Cx1x2 = len(gzip.compress(x1x2.
10             encode()))
11         ncd = (Cx1x2 - min(Cx1, Cx2)) / max(
12             Cx1, Cx2)
13         distance_from_x1.append(ncd)
14 sorted_idx = np.argsort(np.array(
15     distance_from_x1))
16 top_k_class = training_set[sorted_idx
17     [:k], 1]
18 predict_class = max(set(top_k_class),
19     key=top_k_class.count)
```

- Красивая, но страшно медленная реализация
- Статью можно найти здесь:  
<https://aclanthology.org/2023.findings-acl.426/>

# Задача

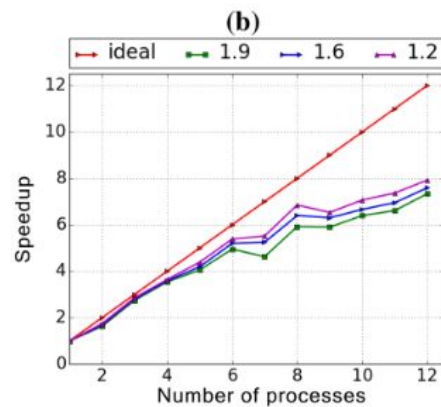
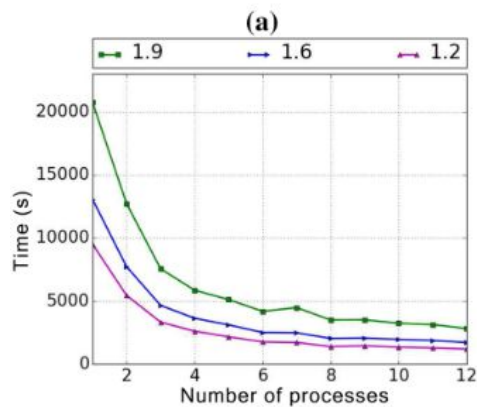
- Пользуясь `numru` и `joblib` ускорить реализацию данного алгоритма;
- `Numru` позволяет ускорить однопоточную реализацию;
- `Joblib` позволяет вам эту реализацию распараллелить.

**Данные:** последовательности  $\Delta RR$ -интервалов извлеченные из ЭКГ и классы ритмов (даны в файлах *X\_train*, *X\_test*, *Y\_train*, *Y\_test*)  
(1 – фибрилляция предсердий, 0 – нормальный ритм)

# Критерии оценивания (максимум 25 баллов)

- **5 баллов** – ускорение с numpy сериальной (однопоточной) реализации (Обратить внимание на типы данных? Может, какие-то из операций можно заменить векторными?)
- **9 баллов** – параллелизация с joblib (На какие независимые задачи можно разбить задачу?)
- **11 баллов** – отчёт о полученной производительности (Сравнение скорости оригинальной реализации со скоростью вашей новой сериальной реализации, ускорение на количестве ядер, графики, разница на тредах и процессах)

# Пример графиков параллелизации



(c)

