Master's Programme Data Science

December 4, 2023

Как отличить шум от хаотического временного ряда

Presenter: Alexander Glushko

Введение

- Временные ряды можно разделить на:
  1. хаотические;
  2. регулярные.
- Как по временному ряду понять к какому классу он относится и отличить его от шума?

Rosso, Osvaldo & Larrondo, Hilda & Martin, M.T. & Plastino, A. & Fuentes, Miguel. (2007). Distinguishing Noise from Chaos. Physical review letters. 99. 154102. 10.1103/PhysRevLett.99.154102.

Мера сложности и энтропия Шеннона

Шенноновская энтропия: $S[P] = -\sum_{i=1}^{N} p_i \ln(p_i)$

Мера на основе Шеноновской энтропии: $H_S[P] = \frac{S[P]}{S_{max}}$

Сложность на основе дивергенция Дженсена-Шеннона:

$$C_{JS}[P] = Q_J[P, P_e] H_S[P]$$

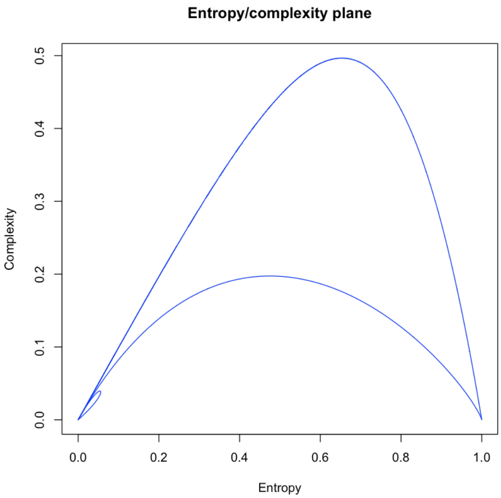
Дивергенция Дженсена-Шеннона:

$$Q_J[P, P_e] = Q_0(S\left[\frac{P + P_e}{2}\right] - \frac{S[P]}{2} - \frac{S[P_e]}{2})$$

Реализация

```r
shannon_entropy <- function(P) {
    P <- P[P != 0]
    return(-sum(P * log(P)))
}

entropic_measure <- function(P) {
    S_max <- shannon_entropy(rep(1, length(P)) / length(P))
    return(shannon_entropy(P) / S_max)
}

jensen_shannon_divergence <- function(P_1, P_2) {
    N <- length(P_1)
    Q_0 <- -2 / (((N + 1) / N * log(N + 1)) - 2 * log(2 * N) + log(N))
    return(Q_0 * (shannon_entropy((P_1 + P_2) / 2) - shannon_entropy(P_1) / 2 - shannon_entropy(P_2) / 2))
}

complexity <- function(P) {
    return(jensen_shannon_divergence(P, rep(1, length(P)) / length(P)) * entropic_measure(P))
}
```

Максимальные и минимальные границы



Реализация

```r
min_borders <- function(d, n_steps = 500) {
    N <- factorial(d)
    d_step <- (1 - 1 / N) / n_steps

    p_min <- seq(1 / N, 1, d_step)
    min_complexity <- numeric(length(p_min))
    min_entropy <- numeric(length(p_min))

    for (i in seq_along(p_min)) {
        P <- c(p_min[i], rep((1 - p_min[i]) / (N - 1), N - 1))
        min_entropy[i] <- entropic_measure(P)
        min_complexity[i] <- complexity(P)
    }

    return(list(min_entropy, min_complexity))
}
```

```r
max_borders <- function(d, n_steps = 500) {
    N <- factorial(d)
    d_step <- (1 - 1 / N) / n_steps

    max_complexity <- numeric(0)
    max_entropy <- numeric(0)

    for (i in seq_len(N - 1)) {
        p_max <- seq(0, 1 / (N - i), d_step)

        for (j in seq_along(p_max)) {
            P <- c(p_max[j], rep((1 - p_max[j]) / (N - i - 1), N - i - 1))

            if (length(P) != N) {
                P <- c(P, rep(0, i))
            }

            max_entropy <- c(max_entropy, entropic_measure(P))
            max_complexity <- c(max_complexity, complexity(P))
        }
    }

    return(list(max_entropy, max_complexity))
}
```

Преобразование временного ряда в вектор

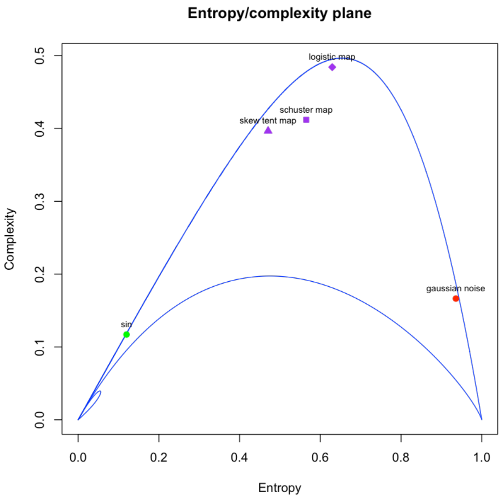
Для того чтобы получить вектор вероятностей из временного ряда воспользуемся порядковым шаблоном.

```r
library(combinat)

ordinal_pattern <- function(x, dim=6) {
    ordinal_numbers <- seq(0, (dim - 1), by = 1)
    possible_pattern <- (combinat::permn(ordinal_numbers))
    result <- 0
    result[1:length(possible_pattern)] <- 0

    for (i in 1:(length(x) - (dim - 1))) {
        temp <- x[i:(i + (dim - 1))]
        tempseq <- seq(0, dim - 1, by = 1)
        tempdata <- data.frame(temp, tempseq)
        tempdata <- tempdata[order(temp), ]

        for (j in 1:length(possible_pattern)) {
            if (all(possible_pattern[[j]] == tempdata$tempseq)) {
                result[j] <- result[j] + 1
            }
        }
    }

    return(result / (length(x) - (dim - 1)))
}
```

Временные ряды для тестирования

- Хаотические:
    1. Skew tent map
    2. Logistic map
    3. Schuster's map
- Гауссовский шум
- Значения синуса

Итоговый график с точками рядов



Thank you!

Repository with all code: https://github.com/Kumokage/entropy_complexity_plane