

# Анализ и прогнозирование временных рядов

Лекция 2. 20.09.2018

$$\lambda_j = \kappa_j - \kappa_{j-1}$$

Общая цель алгоритма Бенеттина заключается в отыскании  $n$  штук ЛНЗ решений линеаризованной системы вдоль некоторой траектории исходной системы  $x(t)$ . При этом в процессе численного интегрирования соответствующих систем ОДУ мы вычисляем так же численно скалярные произведения, фигурирующие в определении многомерных показателей Ляпунова. После произведения достаточного числа шагов интегрирования мы получаем соответствующие оценки значений многомерных показателей ляпунова, а тем самым по вышеприведенной формуле весь ляпуновский спектр.

Дьявол кроется в деталях. Выяснилось, что привычные процедуры численного интегрирования работают плохо, а если быть точным, то вообще не работают.

Ситуация следующая: в силу экспоненциального роста погрешности в одном из направлений, а именно, в направлении, связанном со старшим показателем Ляпунова, выяснилось, что даже если ЛНЗ решения линеаризованной системы  $u_1(0), \dots, u_n(0)$  были ортогональны в начальный момент времени, то они становятся в силу ошибки округления все ближе и ближе друг к другу, а в какой-то момент становятся просто коллинеарными. Более того, коэффициент разложения произвольного вектора по векторам такого базиса ведет себя как  $e^{\lambda_1 t}$ , где  $\lambda_1$  - старший показатель ляпунова и данная величина при  $t \rightarrow \infty$  достаточно быстро выходит за пределы любого машинного типа. В результате оценки полученные таким образом оказались совершенно неадекватными. Решение оказалось элегантным и простым. Поскольку мы имеем дело с лин системой ОДУ, то линейное преобразование соответствующих векторов не меняет значения многомерных показателей ляпунова (**ДЗ** подумать почему так). Соответственно, если мы на некот шаге интегрирования проведем ортогонализацию по грамму-шмидту с последующей нормировкой, то мы получим систему ортонорм вектором, которую используем как начальную для дальнейшего интегрирования линеаризованной системы. Таким образом, алг Бенеттина выглядит:

1. осуществляется линеаризация рассматриваемой ОДУ
2. Задается  $n$  ЛНЗ нач условий  $u_1(0), \dots, u_n(0)$ , это можно сделать, например, выбрав орты пространства  $R^n$ .
3. осуществ совместное интегрирование исходной нелинеаризованной системы  $x = f(x)$  и  $n$  штук линеаризованных систем  $u = A(t)\phi$ , каждая из которых стартует со своих начальных условий. Общее число интегрируемых уравнений составит  $n^2 + n$ , где  $n$  - размерности исходной системы. Для интегрирования разумно использовать метод числ интегр выс порядка точности, к примеру Рунге-Кутты 4-го порядка. В процессе интегрирования вычисляются так же все интегралы, фигурирующие в вычислении многомерных показателей ляпунова. На каждом  $k$ -м шаге интегрирования происходит ортонормировка векторов  $u_1(t_k), \dots, u_n(t_k)$  и поулченные в результате нормировки вектора  $\tilde{u}_1(t_k), \dots, \tilde{u}_n(t_k)$  используются в качестве начальных условий для  $n$  линеаризованных систем. Если вычисл ресурсы позволяют, то  $k = 1$ . В результатемы поулчаем оценки для многомерных показателей Ляпунова, а следовательно и для всего ляпуновского спектра.

**ДЗ** написать на псевдокоде алгоритм Бенеттина

Случаи, когда нам известна системы, которая определяет временной ряд, является исключением из правил и не имеет никакого отношения к реальным системам. В подавляющем большинстве реальных случаев в нашем распоряжении есть только ряд, одно илт многомерный и именно его мы должны протестировать на хаотичность и посчитать горизонт прогнозирования, если он хаотичен. Базовым алгоритмом для решения этой задачи является метод аналогов. идея метода заключается в том, что если мы предполагаем, что переходные процессы в системе завершены и движение происходит в окрестности странного аттрактора, то ряд будет с какой-то периодичностью посещать одни и те же участки странного аттрактора, а это означает две вещи: во-первых, коль скоро траектория посетила одну и ту же область аттрактора, то во временном ряде будут встречаться похожие участки. Если ряд хаотический, то с одной стороны, эти участки будут похожи, а с другой стороны, они буду расходиться как и положено всяким хаотическим траекториям с экспоненциальной скоростью. Оценив скорость расхождения, мы получаем оценку для старшего показателя ляпунова, выполнив эту процедуру и каким-то образом усреднив эти оценки, мы получаем достаточно неплохую оценку для старшего показателя Ляпунова.

### **ДЗ:** посмотреть презентацию

Эти рассуждения приводят к алгоритму, который получил название "метод аналогов", а именно:

- для начального вектора временного ряда мы находим алгоритмом вектор во временном ряде, который наименьшим образом уклоняется от него в смысле выбранной метрики. Если  $x_0$  --- вектор базовой траектории, а его ближайший сосед ---  $\tilde{x}_0$ , то расхождение между двумя траекториями будет равно  $u(t_0) = x_0 - \tilde{x}_0$ , а норма их близости будет просто евклидовой нормой  $\|u(t_0)\|$ . Рассматривая участки временного ряды, стартующие из точек  $x_0$  и  $\tilde{x}_0$  мы будем  $t_0 + 1, t_0 + 2, \dots, t_0 + k_0$  когда  $\|u_0(t_0 + k_0)\| > \varepsilon$ . Связь между ними:  $\|u_0(t_0 + k_0)\| = \|u(t_0)\|e^{\lambda_1 k_0}$ . Логарифмируя, находим оценку  $\lambda_1^{(0)}$ . В точке  $t_0 + k_0$  происходит смена соседа. Мы находим вектор  $\tilde{x}_1$  такой, что cos угла между ним и старым соседом  $\tilde{x}_0(t_0 + k_0)$  максимален. После этого процедура повторяется. Соответственно  $t_0 + k_0 = t_1$  --- точка смены соседа, после чего процедура повторяется снова. Т.о. получаем множество оценок для старшего показателя Ляпунова, которые можно либо усреднить, либо, что лучше, построить линейную регрессию их изменений на соответствующем промежутке времени.

Этот алгоритм является базовым для подавляющего большинства современных алгоритмов вычисления показателей Ляпунова. Основным недостатком является его зависимость от двух параметров, а именно предельной величины малости угла между сменяющимися друг друга соседями и предельной величины расхождения двух соседних траекторий  $\varepsilon$ , когда мы считаем траектории разошедшимися. От подбора этих параметров существенно зависит результат. Современные модификации этого алгоритма две идеи:

1. Не ждем, пока траектории разойдутся, а просто выполняем некоторое малое фиксированное число шагов после отыскания ближайшего соседа. В простейшем случае  $s = 1$ . Этот алгоритм часто реализован в пакетах, иногда его называют алгоритмом Розенштейна.
2. Другая достаточно остроумная модификация носит название метода двойного усреднения. Идея в том, что мы ищем не одного ближайшего соседа при каждой смене, а всех возможных соседей, расстояние до которых от текущей точки основной траектории от текущего ряда не превышает параметра  $\delta$ . Соответственно, если таких соседей оказывается несколько, то мы проделяваем оценку старшего показателя ляпунова многократно с каждым из ближайших соседей. В результате получаем двойное усреднение: 1 - по всем ближайшим соседям, 2 - по всем наблюдениям.

Нужно обратить внимание на то, что алгоритм дает только старший показатель Ляпунова и, следовательно, не дает информации о спектре.

**ДЗ:** Малинецкий и Потапов - алгоритм фрейм-разложения

## Основная формула теории прогнозирования

---

Мы говорили, что при рассмотрении диалектики временных рядов и стоящих ха ними динамических систем мы предполагаем, что переходные процессы завершены и движение происходит в окрестности некоторого аттрактора. В качественной теории ОДУ доказано, что качественно различных типов аттракторов с точностью до выблора системы координат (диффеоморфного преобразования) может быть очень немного.

- Это, во-первых, точка, к которой траектории могут либо экспоненциально сходиться, либо сходиться по спирали. С точки зрения движения это порождает либо затухание, либо затухающие колебания. С точки зрения прогнозирования, если мы развернем эту картину во времени, то получим флуктуации относительно некоторого константного значения.
- Второй тип носит название предельного цикла. В фазовом пространстве мы будем наблюдать некоторую замкнутую кривую и все остальные траектории будут к ней сходиться. Соответствующий тип движения носит название автоколебаний.
- Третий тип носит название многомерного тора

$$T^n = S'_1 \times \dots \times S'_n$$

Соответствующий тип движения называется квазипериодическим и старший показатель Ляпунова для него равен нулю. Для двух предыдущих типов он был строго отрицательным.

- 4 тип - странный аттрактор, для которых старший показатель Ляпунова больше нуля, а движение носит название хаотичного (как и соответствующие временные ряды).

Доказано, что других типов аттракторов не существует.

Математически эти рассуждения формализуются следующим образом: есть некоторый первичный объект  $M^k$  -- многообразие размерности  $k$ , дважды дифференцируемое и для нас оно является первичным объектом. Для регулярных аттракторов - это сам аттрактор, для странных, которые не являются поверхностями ни в каком смысле это некоторая его оболочка. Как только мы задаем систему координат на этом многообразии, мы тем самым сразу каждой его точке ставим в соответствие некую точку из  $R^n$ . Это означает, что мы задаем некоторое отображение  $f: M^k \rightarrow R^n$ . Если это отображение происходит без пересечений, т.е. сохраняет структуру многообразия, то топологи называют такое отображение вложением (embedding). С точки зрения теории прогнозирования важным вопросом является то, при каких условиях наблюдаемый ряд будет сохранять структуру аттрактора. Это достаточно известная в топологии теорема Уитни: при выполнении определенных условий мы гарантированно получаем вложение при выполнении условия  $n \geq 2k + 1$ .

Для того, чтобы уйти от ограничений, накладываемых теоремой Уитни, было предложено рассматривать не временной ряд как таковой, а временной ряд сконкатенированный из  $m$  штук последовательных наблюдений. Построенные таким образом вектора получили название Z-векторов и их размерность равна  $\dim Z = m \cdot s$ , где  $s$  - размерность наблюдений исходного ряда. Т.о., выбирая  $m$  достаточно большим для выполнения условий теоремы Уитни и переходя к рассмотрению ряда из

Z-векторов получаем гарантированное выполнение теоремы Уитни. Это рассуждение формализуется теоремой Такена

Пусть многообразие  $M^k$  как минимум дважды дифференцируемо, его ранг не меньше  $k$ , а собственные значения матрицы линеаризации во всех его особых точках и циклах порядка  $k \leq k! = 1$  различны, то отображение  $M^k$  на пространство  $R^n$  является вложением.

$x_1, \dots, x_n$  --- точки динамической траектории, которая движется по аттрактору  $M^k$ . Эта траектория непрерывна и о ней ничего не известно

$x_{i+1} = g^\tau x_i$ , где  $\tau$  - физическое время, проходящее между двумя наблюдениями нашего ряда и рассмотрим ряд  $y_1, \dots, y_n, \dots$  (тот, который мы наблюдаем) и связь, которая связывает их  $y_i = h(x_i)$  - наблюдаемая. Перейдем в пространство Z-векторов:

$$z_i = (y_i, y_{i+1}, \dots, y_{i+n-1})$$

Для того, чтобы организовать отображение точек многообразия  $M^k$  на пространство Z-векторов выполним следующее:

$$y_i = h(x_i)$$

$$y_{i+1} = h(g^\tau x_i), \dots, y_{i+n-1} = h(g^{\tau(n-1)} x_i)$$

Тогда, объединив эти равенства в одно векторное выражение, связывающее точку на аттракторе  $x_i$  с z-вектором, построенным по наблюдаемому временному ряду, получим

$$z_i = \Lambda x_i$$

Если выполнены условия теоремы Такенса, а с практической точки зрения это означает, что размеры z-векторов достаточно большие, то  $\Lambda$  - взаимно-однозначное, дифференцируемое и более того, обратное отображение  $\Lambda^{-1}$  так же дифференцируемо (диффеоморфизм). из этого факта следует, что наблюдая ряд в пространстве z-векторов, мы наблюдаем истинное движение системы на аттракторе

2. Все свойства. в частности значение старшего показателя Ляпунова, сохраняется. Следовательно, в пространстве z-векторов мы можем оценить характеристики подлинной системы (показ Ляпунова, размерность аттрактора ...)
3. Это соотношение позволяет организовать связь  $z_{i+1} = \Lambda x_{i+1} = \Lambda g^\tau x_i = \Lambda g^\tau \Lambda^{-1} z_i = \Phi(z_i)$ . Выберем только первую компоненту.

$x_{i+1} = \Phi_1(x_i, x_{i-1}, \dots, x_{i-n+1})$  - основная формула теории прогнозирования.

1. Это доказывает, что мы в принципе можем прогнозировать
2. Не нужен бесконечный ряд, только небольшое число предыдущих наблюдений
3. Дифференцируемость = малые изменения в аргументах приводят к малым изменениям функции.  $\Phi_1$  - дифференцируема, а это значит, что не будет дикого поведения ряда.

В классическом прогнозировании знаем структуру функции  $\Phi$ . С одной стороны, в реальных системах можно понять структуру связи между наблюдениями. для реальных моделей существуют маркеры, которые помогают понять, какова реальная структура функции. Есть возможность оценить статистически, насколько мы угадали модель.

Для хаотических рядов структура прогнозной модели настолько сложна, что описать ее с помощью простых функций невозможно. Структуру пытаются оценивать, т.е. вместо  $\Phi_1$  рассматривать  $\hat{\Phi}_1(\theta)$ ,  $\theta \in R^s$  (к примеру, с помощью нейросетей, тогда параметры - веса). Другие подходы ИИ порождают другие способы аппроксимации этой функции, но тем не менее общая идея остается той же. В курсе будем рассматривать прогнозирование на основе кластеризации (predictive clustering, van Raedt & Raimon). Общая идея подхода: если предполагаем выполнены условия теоремы Такенса, то участки странного аттрактора посещаются многократно, что означает, что в хаотическом временном ряде мы будем наблюдать подобные последовательности наблюдений, а соответственно взяв всевозможные последовательности наблюдений всевозможной длины и сгруппировав их с помощью того или иного метода кластеризации, мы можем получить некие характерные последовательности как центры этих кластеров (лейтмотивы, motif, chunk, typical sequence). Соответственно, взяв все возможные лейтмотивы при достаточно большой длине ряда мы можем прогнозировать потом этот ряд, а именно, если среди наблюдений ряда, которая не принимала участия в кластеризации мы наблюдаем и фиксируем наблюдения, которые близки к первой  $n-1$  точке того или иного лейтмотива, то в качестве прогнозного значения для следующей точки ряда мы используем последнюю точку лейтмотива.

*Hint:* не обязательно использовать последовательные наблюдения.