

Анализ и прогнозирование временных рядов

Лекция 3 27.09.2018

Хаотические временные ряды

В рамках парадигмы нелинейной динамики задача определения параметров, фигурирующих в основной формуле носит название задачи выбора параметров реконструкции. Мы реконструируем тот самый странный аттрактор, вокруг которого движется ряд. Есть три параметра:

1. функция h - наблюдаемая - выбор оптимальной наблюдаемой
2. τ - физическое время между наблюдениями.
3. m - размерность z -векторов.

1 и 2 обычно заданы условиями решаемой задачи. И то, и другое определяется физическими или экономическими процессами, которые мы наблюдаем. m - наш произвол.

Для определения m был предложен подход, который получил название определения числа ложных ближайших соседей (false nearest neighbors, FNN). Идея FNN очень проста. Если мы предполагаем, что размерность m выбрана правильно, то если два вектора были соседями в пространстве вложения размерности m $\|x_i - x_j\|$, то с высокой долей вероятности они должны быть соседями в пространстве размерности $m+1$, а те пары векторов, для которых это требование нарушается, носят название ложных ближайших соседей. На понятие FNN можно посмотреть с сугубо аналитической точки зрения. Ф-я ϕ , которая ставит в соответствие предыдущим z -векторам следующие - дифференцируема как минимум два раза, а следовательно, по вполне очевидным представлениям мат анализа, мы имеем, что близость аргументов в подавляющем большинстве случаев должно давать и близость значений. Таким образом, мы естественным образом получаем требование отсутствия FNN. FNN бывают двух видов:

1. FNN - m слишком мало, траектории скомканы, самопересекаются, перепутаны в клубок.
2. FNN на складах (FNN on folds, FNF) - m слишком велико.

Для определения оптимального m для выбора этого параметра реконструкции используется перебор начиная с малых значений. Для каждого значения m оцениваем число FNN и в тот момент, когда их число резко упало, возникает ситуация оптимального m . В некоторых случаях мы можем говорить и о выборе τ . Это значение само по себе не является отдельным параметром и его рассматривают в совокупности с m в виде единого параметра $w = (m - 1)\tau$ - ширины окна реконструкции. Для w имеют место те же самые проблемы, что и для размерности вложения m . Оно м.б. слишком узкими или широким равно с той же аргументации. Однако некоторые эмпирические оценки для этой величины удалось получить. Для удобства рассмотрим непрерывный ряд $x(t)$, в котором окно реконструкции приложено к середине. Будем рассматривать $x(t + \frac{\alpha w}{2})$, где $\alpha \in [-1, 1]$. На этом промежутке рассмотрим дискретные равноотстоящие точки α_s , отвечающие отсчетам, отстоящим на величину τ . Далее разложим в ряд Тейлора:

$$x(t + \alpha w/2) = \sum_{k=0}^N \frac{1}{k!} \left(\frac{\alpha w}{2}\right)^k x^{(k)}(t) + o(w^{N+1})$$

Рассм $c_k = (\alpha_1^k, \dots, \alpha_s^k)$, тогда $z(t) = \sum_{k=1}^N \frac{1}{k!} (w/2)^k x^{(k)}(t) e_k + O(w^{N+1})$

Ортогонализируем e_k , тогда в новом базисе вектор $z(t) = \sum_{k=1}^N A_k \nu_k + O(w^{N+1})$, где ν_k - вектора нового ортонормированного базиса. Поскольку базис ортонормирован, то $A_k = (z(t), \nu_k)$. То, как мы дефинировали вектора e_k , приводит нас к тому, что ν_k - дискретные полиномы Лежандра. Соответственно, заменяя дискретное скалярное произведение его непрерывным аналогом мы получаем

$$A_k \approx \int_{-1}^1 x(t + \alpha w/2) P_k(\alpha) d\alpha$$

$$P_k(\alpha) = \frac{\sqrt{k+0.5}}{2^k k!} d^k / d\alpha^k (1 - \alpha^2)^k$$

Соответственно, раз проинтегрировав интеграл по частям с учетом того, что полиномы обращаются на границах в ноль вместе с производными соответствующего порядка получаем следующее выражение:

$$A_k = (-1)^k \frac{\sqrt{k+0.5}}{2^k k!} (w/2)^k \int_{-1}^1 x^{(k)}(t + \alpha w/2) (1 - \alpha^2)^k d\alpha$$

Воспользовавшись соответствующей теоремой мат анализа получим:

$$= (-1)^k \frac{\sqrt{k+0.5}}{2^k k!} (w/2)^k x^{(k)}(t^*) \int_{-1}^1 (1 - \alpha^2)^k d\alpha$$

Проинтегрировав соответствующие выражения, получаем окончательно

$$= (-1)^k \frac{\sqrt{k+0.5}}{\Gamma(k+3/2)k!} \Gamma(1/2) (w/4)^k x^{(k)}(t^*)$$

Нам удобно считать, что величина k -й производной в точке t^* имеет тот же порядок, что и величина k -й производной в любой точке окна реконструкции. поэтому в этих выкладках можем заменить $x^{(k)}(t^*)$ на $x^{(k)}(t)$. тем самым можем посчитать разброс величин k -й производной от x и соответственно разброс коэффициентов A_k . Было установлено, что для достаточно малых w вектора ν_k ведут себя так же, как базис получаемый в методе главных компонент (базис Каруна-...). Метод главных компонент предназначен для выделения главных направлений с тем, чтобы все остальные направления несли на себе малую информацию. Проекция на все остальные направления близки к нулю - это шум. А вот хаотические данные (это из свойство) не имеют главных направлений. После применения ортогонализации коэффициенты должны оставаться большими. Если это не так, то окно реконструкции выбрано слишком маленьким. Мы скомкали аттрактор и ГК дают информацию о структуре пространства, в которое мы неудачно вложили аттрактор. В качестве эмпирической идеи для выбора ширины окна реконструкции принято считать, что наименьшее значение, при котором порядки коэффициенты первого и второго главных направлений станут равными, можно считать как некоторая своеобразная оценки для w . В наших терминах :

$$\sigma(A_0) = \sigma(A_1) \rightarrow \sqrt{2}\sigma(x(t)) = \frac{w^*}{\sqrt{6}}\sigma(dx/dt)$$

Отсюда получаем грубую оценку на w^* .

В классической теории прогнозирования есть свои способы оценки окна реконструкции w и размерности вложения m . В качестве оценки можно выбирать первый нуль частичной автокорреляционной функции или первый минимум функции взаимной информации.

Соответственно, в хаотических рядах эти концепции **не работают**. Частичная автокорреляционная функция не имеет нуля, а функция взаимной информации минимума. Лбюи ноль и минимум существует, но не имеет отношения к истинному значению окна реконструкции.

Алгоритмы кластеризации, применяемые при прогнозировании временных рядов.

В рамках парадигмы мы должны кластеризовать участки временного ряда, составленные из последовательных или непоследовательных наблюдений. Это означает, что для реальных рядов мы должны проводить весьма значительные вычислительные операции. Долгое время эти задачи считались computationally prohibited. Более того, основной проблемой подавляющего большинства подходов к кластеризации является необходимость предварительного знания числа кластеров. Обычно число кластеров перебирают в разумных пределах. Но в данном случае это невозможно, поэтому возник запрос на алгоритмы, которые не требуют знания числа кластеров, зависят от как можно меньшего числа внятных параметров и, наконец, в идеальном варианте, вообще не зависят от параметров. В качестве парадигмы, которая позволяет решать эти задачи, используется кластеризация с использованием аппарата теории графов и/или community detection. В целом, задача состоит из двух этапов:

1. Превращение данных в граф или в сложную сеть
2. Кластеризация.

Вообще говоря эти этапы независимы и существует множество подходов, которые рассказывают, как данные превратить в граф и существует множество подходов, которые рассказывают, как для данного графа выделить сообщества. Все множество алгоритмов кластеризации составляет собой декартово произведение методов получения графа в данных и методов выделения связанных подмножеств. Тем не менее, бывают алгоритмы, органически сочетающие оба этапа. В целом мы должны определить понятие расстояния, причем мы должны иметь в виду, что обычная евклидова метрика может плохо работать на данных большой размерности (манхэттенская, KL-дивергенция, методы, связанные с современной геометрией и т.п.). Далее мы должны указать меру качества разбиения и алгоритм выделения сильно связанных компонент графа.

Методы превращения данных в граф

(точки выборки отвечают вершинам)

- NNG - nearest neighbor graph - ребром соединяются две ближайших точки.
- Минимальное остовное дерево (MST). Алгоритм Прима, Краскала.
- RNG - relative neighboring graph - вершины соединяются ребром, если не существует третьей вершины, которая ближе к двум этим вершинам.
- Gabriel graph - вершины соединяются ребром, если в гиперсфере вокруг диаметра не содержится других точек выборки.
- Delauney triangulation.

Модифицированный алгоритм Wishart

(Лапко, Ченцов “Непараметрические системы обработки информации”)

Базируется на непараметрической оценке плотности вероятности

$$p_k(x) = \frac{N}{kV_k(x)}$$

где k - параметр, $V_k(x)$ - объем гиперсферы с центром в x и радиусом $d_x(x)$ - минимальный радиус такой, что в внутри гиперсферы этого радиуса содержалось k точек выборки.

В качестве графа данных выбирается граф $U_n = \{\{x_i\}, (x_i, x_j) : d(x_i, x_j) \leq d_k(x_i)\}$.

Выборка сортируется по возрастанию веричин $d_k(x_i)$ и это позволяет ввести подграф U_s , множество вершин которого совпадает с элементами отсортированной выборки до s -го включительно, а множество ребер такое, что обе вершины $\in U_s$. Кроме того, для кластеров вводится понятие значимости, которое определяется следующим образом: кластер значим относительно уровня значимости h , если $\{|p_k(x_i) - p_k(x_j)| \geq h \forall (x_i, x_j)\}$. Параметр h - высота.

В цикле по элементам отсортированной выборки x_i выполняем следующие действия:

1. Если это изолированная вершина графа U_i , то мы формируем из нее отдельный кластер.
2. Если x_i связана только с элементами кластера l_1 и этот кластер еще не сформирован, то мы относим эту вершину к кластеру l_1 ($w(x_i) = l_1$).
3. Если вершина x_i только с эл-ми l_1 , однако этот кластер уже сформирован, то мы полагаем $w(x_i) = 0$, т.е. не относим ее ни к какому кластеру
4. Если x_i связана с кластерами $l_1, \dots, l_t, t > 1$, то
 1. число значимых кластеров $z(h) \leq 1 \vee l_1 \neq 0$, то мы естественно относим $w(x_i) = l_1$ и, более того, присоединяем к этому кластеру вершины всех остальных незначимых кластеров, устанавливая значение w для них равным l_1 .
 2. обратная ситуация - $z(h) > 1 \vee l_1 = 0$, тогда $w(x_i) = 0$, все значимые кластера, связанные с x_i , помечаем как сформированные, а все незначимые кластера уничтожаем, полагая для входящих в них вершин $w(x_j) = 0$.

Алгоритм Zahn

1. Берем полносвязный взвешенный граф, веса - расстояния между точками
2. Строим по нему минимальное остовное дерево (Prim, Kruskal)
3. После чего выкидываем ребра, которые мешают нам жить. Идея заключается в том, что мы пытаемся поделить множество ребер на две части.
 1. "хорошие" ребра, отвечающие вершинам, принадлежащим одному кластеру
 2. "плохие" ребра, соединяющие вершины из разных кластеров.

Когда количество точек небольшое, то можно просто перебрать все ребра и найти некий оптимум некоторой функции качества кластеризации. Однако речь здесь идет о сложных сетях, где это невозможно.

Варианты разделения ребер на две категории:

1. подсчет средней длины ребер в остовном дереве и в качестве границы, отделяющей состоятельные ребра от несостоятельных положить следующую:

$$\lambda \frac{1}{N-1} \sum_{e \in E'} w(e)$$

λ - эмпирический параметр, E' - миним остовное дерево, N - количество вершин. (Fathy-Vagaraeky Aborai)

Такой подход позволяет выделять кластера самой различной формы, однако плохо работает с кластерами различной плотности, что для хаотических рядов крайне существенно.

2. считаем среднее не по всем ребрам, а по ребрам, смежным с данным

$$\lambda \frac{1}{K(e^*)-1} \sum_{e \in S(e^*)} w(e)$$

3. разделение на два класса с помощью простого алгоритма кластеризации (k-means, c-means, ...)
4. жадным образом удаляем ребра в соответствии с выбранным критерием качества разбиения