Neural SDE: phase trajectories of SDE in the action

A Preprint

Papay Ivan
MIPT University
papai.id@phystech.edu

Vladimirov Eduard
MIPT University

2024 год

Данная статья предлагает углубиться в математический аппарат, на котором строится модель Neural SDE. В ней будет рассмотрено, как вычисление фазовых траекторий СДУ обеспечивает качественный прогноз аномалий во временном ряду. Таким образом это предоставит как возможность эффективнее бороться с шумами, так и, в частности, полезный инструмент для упреждения "чёрных лебедей которые могли бы нарушить корректную работу Neural SDE в виду высокой корреляции элементов анализируемой выборки между собой.

Keywords SDE · Stratonovich integral · More

1 Введение

Сбор данных и подготовка их к последующей обработке всегда были одной из важнейших задач машинного обучения. К сожалению не всегда исследователь может гарантировать их целостность и корректность, ведь для тренировки модели чаще всего требуются выборки из тысяч, а то и десятков тысяч элементов - не удивительно, что в данных допускается наличие "шума влияющего на работу обученной модели. Эта задача остаётся актуальной и для временных рядов, то есть данных, индексированных относительно временной координаты. Естественно желание - имея данные для начала временного ряда, проверять: возможно ли продолжить его новыми данными, насколько такое продолжение будет естественно, и не сломает ли это природу текущего временного ряда в стохастическом смысле?

Отсюда и далее мы сконцентрируемся на работе исключительно с временными рядами. В таком наши данные будут представлять из себя данные о некоем дискретном случайном процессе. Дискретном, потому как входная выборка, как множество, точно не будет континуально в силу естественной ограниченности анализируемых данных. Тем не менее корректно будет перейти к непрерывному случайному процессы в силу того факта, что он порождается сигма-алгеброй из конечномерных распределений, которые реально апроксимировать с помощью данных, предоставленных для обучения модели.

Но пока что это всего лишь слова - как именно мы будем апроксимировать искомые распределения? Если бы природа данных была бы строго детерменированной и мы бы изначально имели представление о распределении рассматриваемого случайного процесса, уместно было бы применить метод интерполяции или линейной регрессии. Но в условиях полной неопределённости по отношению и к характеру распределений, как функций, и к её параметрам - нам потребуется что-то другое. Как вариант: апроксимировать ряд дифференциальными уравнениями.

Сама идея использования обыкновенных дифференциальных уравнений("ОДУ"отсюда и далее) далеко не так нова[1], как могло бы показаться на первый взгляд. Так, примерно с 2017-го года она была использована[2] для создания и теоретического обоснования корректности работы модели Neural ODE. Тем не менее, такой метод был всё ещё слаб в робастном смысле: то есть модель легко подпадала под влияние гауссовского шума, а также была уязвима к состязательным атакам. Модель Neural SDE уже строилась на использовании стохастических дифференциальных уравнений("СДУ"отсюда и далее) и была в этом плане эффективнее своего предшественника. Математический аппарат требовался ещё более серьезный, ведь для вычисления решения СДУ без знаний стохастического анализа, исчисления Ито и Стратоновича обойтись было нельзя.

Главной целью данного исследования является построение decision-rejection(принятие-отрицание) критерия корректности той или иной гипотезы о вероятностном распределении входных данных, как некоторого непрерывного случайного процесса. Таким образом для проверки фрагмента временного ряда на наличие аномалий достаточно применить этот критерий для проверки гипотезы о тождественности распределений для конкретного диапазона и для всего остального ряда - разумно будет заключить, что в ряду происходят аномалии, если природа данных в стохастическом смысле резко поменялась.

Вопрос состоит в том: как мы собираемся это делать? Ответ следующий - полагая, что временной ряд порождается определенными конечномерными распределениями, мы сможем приблизить его с помощью стохастических дифференциальных уравнений. То же, разумеется, применимо и к анализируемому диапазону, который требуется проверить на наличие аномалий. Если фазовые траектории полученных дифференциальных уравнений различаются, то есть происходит резкое их возмущение, то очевидно, что в ряду произошла аномалия.

В прошлых работах, связанных с Neural SDE[2,3,4], СДУ использовались только для построения доверительных интервалов для элементов временного ряда. Этот подход в статье предлагается развить посредством использования фазовых траекторий полученных СДУ. Таким образом, можно будет проверять большие массивы данных на корреляцию между собой. В том числе рассматривается конкретная задача - проверить, что два временных ряда обладают одинаковым вероятностным распределением. А именно проверяется соответствие видеоряда готовки еды и ряда данных, полученных с акселерометра, прикреплённого к его руке: ускорения по трём осям x,y,z. Обладают ли эти данные одной и той же природой?

2 Фазовые траектории и Neural SDE

Давайте подытожим: перед нами стоят следующие задачи - применить метод Neural ODE к временному ряду, учесть гауссовский шум и тем самым свести задачу к модели Neural SDE, вычислить фазовые траектории для временных рядов, предварительно свернув два многомерных временных ряда(из видео и из акселерометра) к минимально возможному размеру и, наконец, сравнить полученные фазовые траектории временных рядов по поведению.

2.1 Neural ODE

Изначально Neural ODE был разработан как альтернатива методу остаточных нейронных сетей, состоящих из последовательности скрытых слоёв, значения на каждом из которых подчинялись следующей формуле:

$$h_{n+1} = h_n + f(h_n, w_n), \tag{1}$$

Где h_n - вход n-го слоя и f(h_n, w_n) - нелинейная функция, параметризованная по w_n

Было предложено[5] представление (1) в виде:

$$h_t = h_s + \int_s^t f(h_l, l; w)dl, \tag{2}$$

А вычисление такого дифференциального уравнения уже есть задача для Neural ODE

Algorithm 1 Neural ODE-solver

Require: динамические параметры w, начальное/конечное время t_0, t_1, конечное значение $z(t_1)$, градиент функции потерь в конечной точке $\frac{\delta L}{\delta z(t_1)}$

$s_0 = [z(t_1), \frac{\delta L}{\delta z(t_1)}, 0_{[w]}]$ ▷ Начальное состояние

$[z(t_0), \frac{\delta L}{\delta z(t_1)}, \frac{\delta L}{w}] = ODESolve(s_0, [f(z(t), t, w), -a(t)^T \frac{\delta f}{\delta z}, -a(t)^T \frac{\delta f}{\delta w}], t_1, t_0, w)$

$return \frac{\delta L}{\delta z(t_0)}, \frac{\delta L}{w}$ ▷ Возвращаем градиенты

2.2 Neural SDE

Для учёта шума в наше дифференциальное уравнение следует добавить недетерменированную компоненту, случайную величину. Получится следующее выражение, являющееся интегралом Стратоновича:

$$dX_t^w = h(t, X_t^w; w)dt + \sigma(X_t^w; w)dB_t, \tag{3}$$

Где $B_t = [B_t^1 ... B_t^n]$ - n-мерный Винеровский процесс

2.3 Построение фазовых траекторий по SDE

2.3.1 Ганкелевы матрицы и ODE

В данном случае в силу детерменированности компонент диффура достаточно было бы взять w, который, как мы показали легко считается с помощью соответствующего солвера.

2.3.2 Обработка диффузии для SDE

Как ранее было показано, SDE по сути, так же, как и интеграл Стратоновича, обычная случайная величина. Занулив диффузию и отсемплировав выборку мы тем самым получим траектории характерные для стохастической части диффура. Правда они будут не так сильно выражены чем траектории от детерминированной части. Уместно будет к Ганкелевой матрице из предыдущего пункта прибавить ганкелевы матрицы из этого, домноженные на предельно малый коэффициент.

2.4 Свёртка многомерных временных рядов

Следует использовать Convergent-Cross Mapping(CCM) для того, чтобы привести два анализируемых временных ряда к одному и тому же размеру.

2.5 Компарация фазовых траекторий двух временных рядов

Для сравнения двух фазовых траекторий, как матриц, просто применим метод PCA для выделения главных компонент - и найдем норму Фробениуса разницы этих двух матриц. Если эта разница будет крайне мала, то очевидна схожесть природы двух процессов: иначе - гипотеза под вопросом.

Список литературы

[1] "Neural Ordinary Differential Equations Ricky" T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, David Duvenaud

[2] "Neural SDE: Stabilizing Neural ODE Networks with Stochastic Noise" Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, Cho-Jui Hsieh

[3] "Riemannian Neural SDE: Learning Stochastic Representations on Manifolds" Sung Woo Park , Hyomin Kim , Kyungjae Lee , Junseok Kwon

[4] "Riemannian Diffusion Models" Chin-Wei Huang, Milad Aghajohari, Avishek Joey Bose, Prakash Panangaden, Aaron Courville

[5] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In Advances in Neural Information Processing Systems, pages 6572–6583, 2018.