**ПОСТРОЕНИЕ РЕГРЕССИОННЫХ ЗАВИСИМОСТЕЙ ПРИ НУЛЕВОМ**

**МЕДИАННОМ ЗНАЧЕНИИ ПОМЕХИ**

**ОЦЕНИВАНИЕ ПАРАМЕТРОВ ФУНКЦИОНАЛЬНОЙ ЗАВИСИМОСТИ ПО ЕЕ ПРИБЛИЖЕННЫМ ЗНАЧЕНИЯМ, СОДЕРЖАЩИМ ПОМЕХУ С**

**НУЛЕВЫМ МЕДИАННЫМ ЗНАЧЕНИЕМ**

**Постановка задачи**

Функция является одним из самых известных математических объектов. Одна из задач, имеющая важное практическое значение, заключается в приближенном восстановлении функции по некой информации о ней – детерминированной и (или) статистической (вероятностной). Примером детерминированной информации о функции  являются ограничения на минимально и максимально возможные значения функции  на заданном интервале []. Примером вероятностной информации является закон распределения случайных помех  в приближенных значениях  функции, которая может описывать некоторый физический процесс. На практике значения  в некотором числе  точек  могут быть получены в результате какого-то физического эксперимента. В этом случае задача приближенного восстановления функции  имеет смысл, если функция описывается конечным числом  параметров , неизвестные точные значения которой будем обозначать как .

Таким образом, речь в нашей работе пойдет об оценке параметров функции

, **R**, [] , (1)

по ее приближенным значениям

 ,  , (2)

когда дополнительно известно, что: 1. вектор  принадлежит заданному ограниченному множеству , например, параллелепипеду в **R**; 2.  – непрерывная случайная величина, значения которой ограничены; 3. ее медиана  равна нулю.

Судя по аннотациям к научным работам, которые удалось просмотреть, наиболее распространенной является линейная модель изучаемой зависимости

, (3)

в частности, полиномиальная, когда

; , . (4)

На практике часто возникает и более сложная задача, где помимо оценивания параметров  необходимо определить сам вид (структуру) функциональной зависимости. Иначе говоря, задается конечное число альтернатив

, **R**, , (5)

и надо определить, к какому из  классов функций  принадлежит функция  и собственно оценить вектор  ее параметров. В школьной программе мы встречались с подобной задачей, когда необходимо было узнать, имеем мы дело с линейной или экспоненциальной зависимостью. Но там задача решалась по *точным* значениям функции.

Задачам приближенного восстановления (описания) функции по ее приближенным значениям (на практике – по экспериментальным данным) посвящены сотни, а, быть может, тысячи научных статей. Обычно при этом используется предположение о равенстве нулю математического ожидания помехи [?,?,?]. "Фишка" нашего небольшого исследования заключается в использовании условия  вместо традиционного условия  равенства нулю математического ожидания случайной величины  , когда алгоритм оценивания параметров функции базируется на идеях метода наименьших квадратов.

Наш интерес к условию  обусловлен тем, что условие  заведомо не выполняется, если измерения ведутся вблизи одной из естественных границ измеряемой физической величины. В этом случае большими по абсолютной величине могут быть лишь помехи *одного знака*. Говоря о помехах мы имеем в виду не только собственно ошибки измерений прибора (тут можно было бы еще спорить, что современная измерительная техника все же является довольно точной), но и влияние каких-то факторов, не учтенных в функции . И хотя каждое из двух условий –  и – не является частным слчаем другого. Можно сказать, что с точки зрения решения практических задач условие  является более слабым. По сути оно означает лишь, что с равной вероятностью 0,5 помеха может быть либо положительного, либо отрицательного знака. Другими словами, условия физического эксперимента не создают условий для преобладания помех какого-то одного знака. И никаких ограничений на абсолютные значения помех! Условие  допускает, что среди реализаций случайной величины  могут встречаться довольно большие по абсолютной величине значения, тогда как среди реализаций другого знака – нет.

Мы ставим перед собой две задачи: 1. "задачу-максимум" – дать алгоритм, способный при определенном уровне помех: идентифицировать из конечного числа альтернатив класс функций, к которому принадлежит изучаемая зависимость; достаточно точно оценить ее параметры; достаточно надежно оценить точность полученных приближенных значений неизвестных параметров функции; 2. "задачу-минимум" – дать алгоритм, способный: довольно надежно оценить параметры функциональной зависимости, тип которой известен; опять же, оценить точность восстановления неизвестных параметров функции. В качестве основного критерия при поиске оптимального решения мы будем использовать оценку математического ожидания погрешности приближенных оценок истинных параметров изучаемой функциональной зависимости.

Как известно, практика – критерий истины. Ясно, что качество решения нашей задачи зависит от ряда факторов, в числе которых: 1. соотношение между числом  измерений  и числом  оцениваемых параметров , 2. интенсивность помех , 3. числом  альтернатив и, что более важно, степенью близости классов функций . А это значит, что для подтверждения теоретических выкладок необходим достаточно широкий вычислительный эксперимент. Мы выполним необходимые расчеты с помощью собственного программного обеспечения.

**Алгоритм**

Задача сглаживания с условием  уже рассматривалась в математике. Считается [?], что в этом случае надо минимизировать сумму модулей отклонений модельной зависимости от точек . Но при этом нет ссылок на то, что можно, пусть и грубо, оценить уклонение построенной модельной зависимости  от неизвестной истинной . А может ли быть большой смысл от приближенного решения задачи, если нет никакой оценки его точности? Метод минимизации модулей также не предполагает наличие априорных ограничений на вектор . Как быть, если вектор параметров , обеспечивающий минимум суммы модулей, не принадлежит множеству ?

Понятно, что каждом отдельно взятом случае фактическая точность решения модельного примера (когда истина известна) не может служить ни сравнительной оценкой двух конкурирующих алгоритмов, ни критерием эффективности любого отдельно взятого алгоритма. Заведомо ясно, если все, или почти все, помехи имеют один знак (а условие , как, впрочем, и условие , это допускают, пусть и с маленькой вероятностью), то ни один метод не сможет дать хороший результат. И также, при определенном "раскладе" помех  теоретически более обоснованный метод может по воле случая уступить менее состоятельному методу. При оценке эффективности метода надо ориентироваться на средние результаты по некоторому множеству ситуаций и примеров. В связи со сказанным, наша идея заключается в том, чтобы за качество построенного приближения  к  взять оценку математического ожидания

  (6)

близости (расстояния)  функции  от , где в роли расстояния можно, к примеру, взять функции

, (7)

, (8)

. (9)

При решении задачи, которую мы обозначили выше как "задачу-минимум", подход с критерием (6) рассматривался в работе [?]. Забегая вперед, скажем, что мы предложим более конструктивный, чем в [?], алгоритм. Подчеркнем также, что наша "задача-максимум" в указанной работе не рассматривалась.

Фигурирующую в -кратном интеграле (6) функцию плотности вероятностей , , где  – вектор, который мог бы быть неизвестным истинным вектором , можно построить на основе формулы биномиального распределения случайной величины [?]. В самом деле, пусть:  – один из векторов, претендующий на то, что именно он и является неизвестным вектором  функции (3);  – элементы последовательности

, ,...,; (10)

 – дискретная случайная величина, принимающая значения

, (11)

где

. (12)

Очевидно, что в содержательном плане значение  есть число переходов знаков элементов последовательности (10) и этим значением могут быть натуральные числа от 0 до . Если бы действительно случилось, что , то значения  были бы ничем иным, как помехами , и в силу условия  можно было бы записать вероятности  событий :

 ,  . (13)

Перейдем от обсуждения вопроса с числом переходов знаков при каком-то одном векторе  к анализу этой ситуации по отношению ко всему множеству . Можно сказать следующее. Существует некоторое разбиение множества  на систему из  подмножеств  таких, что элементы (вектора ) подмножества  обеспечивают одно и тоже число  переходов знаков элементов в последовательности (10). Соответственно, априорная вероятность того, что неизвестный истинный вектор , вычисляется по формуле . Но в каждом конкретном случае некоторые из множеств  могут оказаться пустыми, то есть ни при каком векторе параметров  число переходов знаков элементов последовательности (10) не будет равным данному  (любой желающий может в этом убедиться, попытавшись провести прямую, которая бы пересекла наугад нарисованную ломаную линию так, чтобы все соседние вершины ломаной оказались по разные стороны от прямой). Априорные вероятности  попадания неизвестного вектора  в непустые множества  можно пересчитать в апостериорные . Обозначим через  множество номеров всех непустых подмножеств . Тогда

 , . (14)

Поскольку у нас компоненты вектора , отвечающего за возможные "состояния" истинного вектора , величины непрерывные, а не дискретные, то чтобы применить формулу (6), необходимо от оценок  вероятностей событий  перейти к оценкам , , функции плотности вероятностей. У нас нет никакой информации, которая позволила бы как-то ранжировать предпочтению вектора  в пределах каждого из подмножеств . В подобных случаях логично взять равномерное распределение

 , , (15)

где – мера (аналог величины объема) подмножества  в пространстве **R**. После этого формула (6) приобретает вид

, (16)

где – лаконичная запись многомерного (у нас, -мерного) интеграла. В общем случае, как наш, пределы интегрирования каждого одномерного интеграла зависят от переменных, по которым осуществляется интегрирование в каждом одномерном интеграле, внешнем по отношению к данному [?]. Это существенно усложняет вычисление многомерных интегралов. Опять же, забегая вперед, заметим, что предлагаемом нами алгоритме интегрирование осуществляется по многомерным параллелепипедам, когда пределы интегрирования каждого интеграла являются постоянными.

Пусть в (16) расстояние  берется в виде (8). Тогда имеет место цепочка преобразований.

 

 

 

. (17)

И окончательно:

. (18)

Чтобы найти вектор , который доставит минимум  функции (17) и который мы возьмем в качестве наилучшей оптимальной оценки неизвестного вектора , необходимо (по аналогии со случаем функции одной переменной) взять первые производные  функции (18) по каждой из переменных  [?]. После этого, приравнять эти производные к нулю и решить полученную систему линейных уравнений. Мы имеем дело со случаем, когда система распадется на  отдельных линейных уравнений с одной переменной:

  (19)

Эти уравнения имеют следующие решения:

 , , (20)

где

 . (21)

Формулы (20)-(21) – пусть в несколько иных обозначениях – представляют собой решение поставленной задачи по алгоритму, предложенному в [?]. С точки зрения практической реализации алгоритма, базирующегося на формулах (20), (21), возникают два вопроса: 1. Можно ли дать простой способ построения множеств ; 2. Можно ли дать простой способ вычислении интегралов по этим множествам, которые фигурируют в правой части формулы (21). Выполненные нами геометрические построения в случае  и числа точек  порядка 10 показали, что области  имеют достаточно сложную геометрию (некие звездообразные фигуры со многими лучами) (рис. 1). Однако, проблему можно решить по принципу "гордиева узла" – отказаться от работы непосредственно с множествами , а использовать их точечную (сеточную) аппроксимацию (причем без того, чтобы описать эти множества явно) и воспользоваться аналогом формулы (20).

Суть предложения состоит в следующем. Зададимся достаточно большими натуральными  и -мерным параллелепипедом

R, (22)

содержащим множество  (если множество  само является параллелепипедом, то ). Покроем этот параллелепипед достаточно густой сеткой  с

 (23)

узлами , , где каждая координата , , может независимо от остальных () координат принимать одно из  значений

, . (24)

Параллельно с сеткой  введем систему  параллелепипедов R, , центрами которых являются ее узлы и стороны которых равны

: (25)

. (26)

Эти параллелепипеды образуют покрытие (замощение) параллелепипеда . Отберем из названных  узлов те, что принадлежат множеству . Пусть  – число отобранных узлов. Перенумеруем отобранные узлы, отдав им первые  номеров .

Для каждого  найдем элементы , , последовательности (10) и по их значениям, используя формулу (11), вычислим число переходов знаков элементов последовательности (10). Пусть ,  – множество номеров  отобранных узлов , обеспечивающих  переходов знака, и  – число таких узлов. Таким образом, не прибегая непосредственно к построению подмножеств , мы будем аппроксимировать каждое из них объединением

. (27)

Понятно, что предположение о том, что все вектора , принадлежащие параллелепипеду , обеспечат то же число переходов знака, что и центр  параллелепипеда, будет неверно лишь для тех параллелепипедов , которые не принадлежат (целиком) какому-то одному из множеств . Но при густой сетке  и, соответственно, достаточно малых размерах параллелепипедов  процент таких "спорных" параллелепипед от их общего числа  будет весьма мал (так, при  "спорные" параллелепипеды располагаются по линиям, а "не спорные" – по площадям). К тому же, ошибки от отнесения "спорных" множеств  целиком к некоторому множеству , будут (частично) взаимно погашаться. Суммируя все эти доводы, можно сказать, что аппроксимация множеств  объединениями достаточно "мелких" параллелепипедов даст достаточно хорошее приближение к математическому ожиданию (16) ошибки оценивания параметров изучаемой функциональной зависимости:



 (28)

****

****

Если в многократном интеграле пределы интегрирования по каждой из переменных не зависят от других переменных, то такой интеграл равен произведению входящих в него однократных интегралов [?]. Если обозначить координаты граней параллелепипеда  как  и , то легко убедиться, что:

 . (29)

Если к тому же принять во внимание, что

 (30)

 (31)

, (32)

то цепочку преобразований (28) можно продолжить.

 =

****

****

**** (33)

Чтобы найти вектор параметров , который доставит минимум оценке (33) математического ожидания ошибки решения задачи, поступим также, как и в случае с минимумом функционала (18). С учетом того, что последнее, третье слагаемое в (33) не содержит оптимизируемые переменные  и производная по нему равна нулю, находим:

 . (34)

Приравнивая частные производные (34) к нулю получим окончательно:

,  (35)

Формулы (35) и будем использовать в компьютерных программах для проведения вычислительного эксперимента.

А теперь сравним формулы (20), (21) с преложенной нами формулой (35). Превосходство последних очевидно. Прежде всего, нам удалось избавиться от необходимости фактического построения множеств , которые являются, вообще говоря, невыпуклыми многогранниками в **R**. Соответственно, исчезла необходимость вычисления многомерных интегралов по этим множествам. К этому подмешивается еще одна проблема: даже многоугольник на плоскости, не говоря уже о многомерном многораннике, однозначно не определяется координатами своих вершин – необходимо задавать порядок обхода этих вершин [?]. Как это делать без визуализации и когда таких многогранников много? Мы же для построения оптимальных оценок  нуждаемся лишь в координатах точек регулярного покрытия множества , для узлов которого необходимо лишь вычислить число переходов знаков элементов последовательности (10).

**Результаты вычислительного эксперимента**

**Заключение (Выводы)**

Мы полагаем, что лучшее, что можно поместить в заключение работы, это очертить круг тех условий, при которых мы могли бы советовать использовать предложенный нами алгоритм. Понятно, что строго очертить этот круг невозможно уже хотя бы в силу ограниченного объема численных расчетов. Однако, приведенные рисунки и таблицы достаточно наглядно свидетелствуют о следующем.

**ПОЯСНЕНИЯ ОТ 14.09.2016 К ДАЛЬНЕЙШЕЙ ДОРАБОТКЕ ТЕКСТА**

После первых удачных расчетов, когда стало ясно, что, в принципе, мы

выбрали подходящую тему, можно составить план дальнейших работ.

**Какие аспекты проблемы можно было бы еще рассмотреть:**

1. Выбор в качестве наилучших оценок один из тех векторов параметров, что дает наибольшее число переходов знака (ссылаясь на то (и привести тому в подтверждение рисунок), что в этом случае график модельной функции должен прижиматься к графику неизвестной истинной функции). И этот феномен возможен именно при равенстве нулю медианы помехи.
2. Попытаться скомбинировать наш метод с принципом доверительной вероятности (может удасться сблизить значения математического ожидания точности решения с собственно фактической точностью решения).
3. Если при решении проблемы структурной идентификации функциональной зависимости вычисленная оценка математического ожидания ошибки решения будет все-таки существенно отличаться от фактической точности, то это еще не катастрофа. Важно, чтобы значение математического ожидания надежно указывало на правильный тип функции при не очень высоком уровне помех в приближенных значениях функции.
4. В отношении многовариантных расчетов сказать, что ты нигде не читал, но думаешь, что это уже высказывали другие, что если 10 или 20 примеров убеждают в пользу алгоритма, какими бы эвристиками он ни обладал, то психология человека такова, что он поверит этому алгоритму, если даже высокая теория это не подтверждает. И наоборот, если теория убеждает в оптимальности какого-то метода, а тот "облажался" подряд 10 раз то никто не поверит в этот алгоритм.

**Что нужно сделать в теории:**

1. Показать на примере линейной функции  при 8-10 измерениях, сколь *мала (в предположении* может быть область параметров , вероятность принадлежности которой истинного вектора параметров  при этом достаточно *высока*. И вот это и предопределяет успех. Эта область, кстати, представляет звездообразный многоугольник с многими лучами.

**Что нужно сделать в плане программирования:**

1. Пусть не сейчас, а ближе к моменту написания окончательного варианта работы, запрограммировать в **Программе 1** запись нужных файлов. Чтобы потом было легче строить таблицы и рисунки.

**Какие примеры следовало бы посчитать**:

1. По **Программе 1** серию примеров при разном уровне помех, серию примеров при различном числе точек задания приближенных значений функции. Причем под каждым примером мы понимаем, в свою очередь, серию расчетов с различными помехами.
2. По **Программе 2** посчитать один пример с различными помехами.
3. По **Программе 3** посчитать один пример при различном уровне помех. Будет здорово, если мы будем наблюдать следующее: с уменьшением помехи алгоритм все более надежно определяет вид функции.