

**Министерство образования и науки РФ
ФГАОУ ВПО «Казанский (Приволжский) федеральный университет»
Институт вычислительной математики и информационных технологий
Кафедра системного анализа и информационных технологий**

Р.Х. Латыпов, Е.Л. Столов

**Биометрические способы идентификации личности
в системах информационной безопасности**

Учебно-методическое пособие
Предназначено для студентов 4 курса Института
вычислительной математики и информационных технологий

Казань 2014

УДК 511, 519.6

Печатается по решению редакционно-издательского совета ФГАОУВПО
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»,
методической комиссии факультета вычислительной
математики и кибернетики, протокол № 4 от 11 декабря 2014 г.

Авторы-составители –
докт. техн наук, проф. каф. САИТ КФУ Р.Х. Латыпов
докт. техн наук, проф. каф. САИТ КФУ Е.Л.Столлов

Рецензент –
докт. физ.-мат. наук, проф. Ш.Т.Ишмухаметов

Р.Х. Латыпов, Е.Л.Столлов

Биометрические способы идентификации личности: учебное пособие / Р.Х.
Латыпов, Е.Л.Столлов.– Казань: Казан. ун. 2014.– 29 с.

Предназначено для студентов старших курсов факультета вычислительной математики и кибернетики.

©Казанский университет, 2014

Содержание

1	Введение	4
2	Построение классификаторов	5
2.1	Основные проблемы в задаче идентификации	5
2.2	Линейная классификация	6
2.2.1	Разбиение на два класса. Персептрон	6
2.2.2	Произвольное конечное число классов	8
3	Линейные классификаторы 2	9
3.1	Метод наименьших квадратов	9
3.2	Метод Фишера для двух классов	10
3.3	Метод Фишера для произвольного числа классов	11
4	Классификация на основе метода Байеса	11
4.1	Классический подход	11
5	Нейронные сети	12
5.1	Тренировка нейронной сети	12
5.2	Практическое вычисление весов	14
5.3	Метод деформируемого многогранника в задаче тренировки нейронной сети	14
5.4	Нейронная сеть на основе линейной аппроксимации	15
6	Кластеризация	16
6.1	Алгоритм K-means	17
7	Радиальные сети	18
7.1	Постановка задачи	18
7.2	Реализация на основе локализованного базиса	19
8	Методы понижения размерности	19
8.1	Разложение по ортонормированному базису	20
8.1.1	Дискретное преобразование Фурье (DFT)	20
8.1.2	Дискретное преобразование Хартли (DHT)	20
8.1.3	Дискретное косинус преобразование (DCT)	21
8.1.4	Преобразование Адамара (Hadamard transform)	21
8.1.5	Базис Хотелинга-Карунена-Лоэва	22
8.2	Аппроксимация с помощью произвольного множества векторов	23
8.3	Вейвлет преобразование	24
8.3.1	Непрерывное вейвлет преобразование	24
8.3.2	Дискретное вейвлет преобразование (DWT)	25
9	Сжатое представление последовательностей произвольной длины	28

1 Введение

Один из разделов компьютерной безопасности связан с доступом персонала в то или иное помещение или к отдельным терминалам, а также санкционирование отдельных действий. Система паролей остается наиболее простым и распространенным способом ограничения доступа, но у нее есть свои недостатки. Более интересным является метод обеспечения доступа, основанный на биометрических параметрах личности, который является более универсальным, хотя его применение требует дополнительного оборудования. Известны следующие способы идентификации, использующие биометрические параметры

1. Идентификация по лицу
2. Идентификации по отпечатки пальцев
3. Идентификация по капиллярным линиям ладони
4. Идентификация по радужной оболочке
5. Идентификация по голосу
6. Идентификация по почерку
7. Идентификация по походке

Данный список не является исчерпывающим. С развитием новых технологий появляются совершенно неожиданные способы идентификации. В шпионских фильмах герой фотографирует отпечаток пальца человека и предъявляет его системе. Способ идентификации по капиллярным линиям ладони появился сравнительно недавно, и пока не известен метод его компрометации, поскольку он работает только с живой тканью.

В отличие от идентификации по паролю идентификация по биометрическим параметрам носит вероятностный характер. Идентификация по паролю гарантирует правильность набора пароля, а не легальность использующей его личности, поэтому и здесь правильность идентификации носит вероятностный характер. При идентификации по биометрическим признакам выделяется набор параметров, обладающих достаточной устойчивостью в течение определенного времени. С помощью некоторой решающей функции сравнивается найденный набор с имеющимся в базе, после чего принимается решение о правильности идентификации. Указанный момент является общим для всех рассматриваемых методов, поэтому с математической точки зрения они мало отличаются друг от друга. Основная проблема заключается в выделении нужных параметров, способов их вычисления и доказательства относительной стабильности указанных параметров и их достаточности для получения заданной вероятности идентификации. В результате все рассматриваемые методы носят эмпирический характер, и все оценки правильности идентификации получают статистическими методами. В этой связи в данном пособии собраны математические мето-



Рис. 1: Фотографии из базы a) и фото для сравнения b)

ды классификации. Приложения к конкретным задачам предполагается рассмотреть позже.

2 Построение классификаторов

2.1 Основные проблемы в задаче идентификации

Проблемы, сопутствующие биометрической идентификации, рассмотрим на примере идентификации по фотографии. На Рис. 1 представлены примеры фотографии из базы и фотографии, по которой нужно осуществить сравнение. Очевидно, что прямое сравнение фотографий не имеет смысла. Первая проблема заключается в том, что одна фотграфия сделана анфас, а другая — когда голова повернута. Кроме того, размеры лица на фотографиях сильно разнятся. Для сравнения необходимо привести вторую фотографию к некоторому стандартному виду. Далее будет сказано, каким образом это можно сделать, хотя абсолютного равенства всех условий фотографирования и освещенности добиться не удастся.

Каждая растровая фотография может рассматриваться как набор строк. Однако даже после выравнивания фотографий сравнение соответствующих строк не имеет смысла из-за разницы в освещенности одинаковых точек лица на разных фотографиях. Необходимо извлечь из этих строк информацию, не зависящую от указанных обстоятельств. Если растровая фотография имеет размеры $m \times n$, то мы имеем дело с mn пикселями, каждый из которых является вектором длины 3. Первый шаг в задаче распознавания заключается в уменьшении размерности объекта. Цветное изображение заменяется тремя тоновыми изображениями, каждое из которых изучается независимо от других, однако, на последнем этапе результаты объединяются. Даже тоновое изображение часто содержит избыточную информацию, и его заменяют бинарным изображением. Этот этап сжатия не является последним, возможны и дальнейшие шаги, направленные на уменьшение размерности задачи. Более подроб-

но о методах сжатия будет сказано в последующих главах.

На каком-то шаге фотографиям будут отвечать векторы одинаковой размерности, и нужно будет отнести полученные векторы к одному или к разным классам. Это делается с помощью дискриминантных функций, к изучению которых мы переходим. Задача идентификации разбивается на два этапа:

1. получение тренировочного множества и построение с его помощью дискриминантной функции;
2. принятие решения для заданного вектора по значениям построенной функции.

Если выбор дискриминантной функции уже сделан, то первый этап сводится к оценке параметров этой функции. Что касается второго этапа, то здесь существуют ситуации, когда возможен отказ от принятия решения.

2.2 Линейная классификация

При изложении данного пункта будем следовать книге [1]. Простейшей дискриминантной функцией является линейная функция или набор таких функций. Пусть задано множество векторов размерности d . Любая гиперплоскость в пространстве указанной размерности задается уравнением

$$f(\alpha) = f(a_1, \dots, a_d) = \sum_{i=1}^d w_i a_i + w_0 = 0$$

Коэффициенты $w_i, i = 1, \dots, d$ назовем весами, а коэффициент w_0 – сдвигом. В дальнейшем все векторы считаются столбцами. Положим $\omega^T = \langle w_1, \dots, w_d \rangle$, $\alpha^T = \langle a_1, \dots, a_d \rangle$. Указанное уравнение запишем в форме

$$\omega^T \alpha + w_0 = 0 \tag{1}$$

Эта гиперплоскость разбивает пространство на две части, определяемые неравенствами $f(\alpha) > 0$ и $f(\alpha) < 0$.

2.2.1 Разбиение на два класса. Персептрон

Предположим сначала, что все множество векторов C разбито на 2 класса C_1, C_2 . Требуется построить такую дискриминантную функцию, что $f(\alpha) > 0$, если $\alpha \in C_1$ и $f(\alpha) < 0$, если $\alpha \in C_2$. В случае равенства происходит отказ от решения. Уже в этом случае возможны проблемы. Рассмотрим пример, представленный на Рис.2. Здесь четыре точки, принадлежащие двум классам. Очевидно, что в этом случае не существует линейной дискриминантной функции для данного множества векторов. В тех случаях, когда решение существует, оно может быть найдено с помощью алгоритма персептона. Модифицируем уравнение (1). Сохраним обозначение ω для

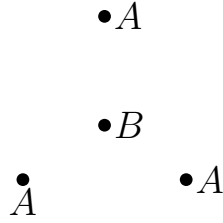


Рис. 2: Пример невозможного разбиения на два класса линейной функцией

вектора $\langle w_0, w_1, \dots, w_d \rangle$, обозначение α для вектора $\langle 1, a_1, \dots, a_d \rangle$. В этом случае уравнение (1) примет форму

$$\omega^T \alpha = 0$$

Предположим, что существует линейная дискриминантная функция. Это означает, что существует такой вектор ω_0 , для которого справедливы неравенства

$$\omega_0^T \alpha > 0, \alpha \in C_1, \quad \omega_0^T \beta < 0, \beta \in C_2$$

Заменяя β на $-\beta$, сведем задачу к отысканию вектора ω_0 , для которого выполнены неравенства $\omega_0^T \gamma > 0$ для всех точек $\gamma \in C$. Поскольку множество C предполагается конечным, последнему условию можно придать вид

$$\omega_0^T \gamma > B > 0 \quad \forall \gamma \in C, \quad (2)$$

где B – некоторое число. Как было показано выше, вектор ω_0 существует не всегда. Если же он существует, то его можно найти с помощью конечного числа итераций. Этот алгоритм получил название алгоритмом обучения персептрона – простейшей нейронной сети.

Теорема 1 *Если существует линейная дискриминантная функция, то она может быть найдена за конечное число шагов алгоритма обучения*

Доказательство. Выберем в качестве вектора $\omega^{(1)}$ произвольный ненулевой вектор подходящей длины. Если вектор $\omega^{(k)}$ уже найден, и существует вектор γ_k , для которого выполнено неравенство $\omega^{(k)T} \gamma_k < B$, то $\omega^{(k+1)} = \omega^{(k)} + \gamma_k$. Алгоритм останавливается, если на каком-то шаге получен вектор $\omega_0 = \omega^{(k)}$, для которого выполнено (2). Допустим, что у задачи имеется решение, но на каждом шаге с номером k находится такой вектор γ_k , для которого имеет место неравенство

$$\omega^{(k)T} \gamma_k < B, k = 1, 2, \dots \quad (3)$$

Имеем

$$|\omega^{(k+1)}| |\omega_0| \geq |(\omega^{(k+1)}, \omega_0)| = |(\omega^{(1)} + \gamma_1 + \dots + \gamma_k, \omega_0)| > kB - |(\omega^{(1)}, \omega_0)|.$$

Отсюда следует

$$|\omega^{(k+1)}| > \frac{kB - D}{|\omega_0|} \quad (4)$$

где D некоторое число. С другой стороны,

$$|\omega^{(k+1)}|^2 = |\omega^{(k)}|^2 + |\gamma_k|^2 + 2(\omega^{(k)}, \gamma_k)$$

или

$$\begin{aligned} |\omega^{(k+1)}|^2 - |\omega^{(k)}|^2 &< |\gamma_k|^2 + 2B \\ |\omega^{(k)}|^2 - |\omega^{(k-1)}|^2 &< |\gamma_{k-1}|^2 + 2B \end{aligned}$$

Складывая найденные неравенства, получим, что

$$|\omega^{(k+1)}|^2 - |\omega^{(1)}|^2 < (F + 2B)k,$$

где $F = \max |\gamma_i|^2$. Это означает, что

$$|\omega^{(k+1)}| < \sqrt{(F + 2B)k + |\omega^{(1)}|^2} \quad (5)$$

Последнее неравенство противоречит (4) при больших значениях k , что завершает доказательство.

Из геометрических соображений следует, что в качестве начального вектора $\omega^{(1)}$ следует выбирать вектор пропорциональный сумме векторов γ_i . Сравнение неравенств (5,4) дает оценку числа итераций, когда следует остановить процедуру, если решение еще не найдено.

Если задача разделения двух множеств имеет решение, то произвольный вектор γ отнесем к первому множеству, если $\omega_0^T \gamma > B$, и ко второму множеству, если $\omega_0^T \gamma < -B$. Если ни одно из неравенств не имеет места, происходит отказ от принятия решения. Иногда ставится условие, чтобы решение принималось в любом случае. Это можно сделать, определив, к какому неравенству оказалось ближе полученное значение. При этом следует указать на возникшее обстоятельство.

2.2.2 Произвольное конечное число классов

Пусть тренировочное множество векторов $C = \bigcup_{i=1}^N C_i$ есть объединение непересекающихся подмножеств. Пусть для каждой пары подмножеств $C_i, C_j, i < j$ удалось построить векторы ω_{ij} и числа B_{ij} таким образом, что $\omega_{ij}^T \gamma < -B_{ij}$ для $\gamma \in C_i$ и $\omega_{ij}^T \gamma > B_{ij}$ для $\gamma \in C_j$. Определим дискриминантную функцию $dis_{ij}(\gamma)$ равную -1 в первом случае и 1 во втором. Если ни одно из неравенств не выполнено, функция не определена. Классификация произвольного вектора γ осуществляется следующим образом. Составим матрицу Sol размера $N \times N$, где $Sol[i|i] = 0$, $Sol[i|j] = dis_{ij}(\gamma), i < j$, $Sol[i|j] = Sol[j|i]$. Пусть определены все элементы этой матрицы. Ищем строку с номером j вида $1, \dots, 1, 0, -1, \dots - 1$. Назовем такую строку правильной. Если эта строка правильная, то вектор относим к классу C_j , если правильные строки отсутствуют, то отказываются от решения. Нужно только показать, что не существует

двух правильных строк в одной матрице. Действительно, если есть две правильные строки с номерами j, k , $j < k$. Это означает, что $Sol[j|k] = -1$, $Sol[k|j] = 1$, но это противоречит определению матрицы.

3 Линейные классификаторы 2

Выше был рассмотрен персептрон, применимость которого ограничивалась случаем линейной сепарабельности любых двух множеств. Даже при наличии указанного условия мы сталкиваемся со случаем отказа от принятия решения. Ниже предлагаются подходы к решению задачи свободные от этих ограничений.

3.1 Метод наименьших квадратов

При изложении данного пункта будем следовать книге [1] Пусть имеется тренировочное множество векторов C , состоящее из N векторов и разбитое на K непересекающихся подмножеств: $C = \cup C_i$. Все векторы имеют одну и ту же длину n . Без ограничения общности можем считать, что первая компонента каждого вектора равна 1. Каждому вектору $\gamma_i \in C_j$ сопоставим вектор μ_i длины K , в котором все компоненты, кроме j -ой, нулевые, а j -ая компонента равна 1. Составим из всех столбцов γ матрицу X , из всех столбцов μ матрицу M , причем вектор γ_i и соответствующий ему вектор μ_i занимают в своих матрицах одинаковые позиции. Требуется построить матрицу W размера $K \times n$ такую, чтобы

$$tr((M - WX)(M - WX)^T) \rightarrow \min \quad (6)$$

Содержательно равенство (6) означает следующее. Требуется построить такую матрицу W , чтобы после умножения на нее всех векторов из класса C_j получался вектор близкий к μ_j . Это обычная задача метода наименьших квадратов. Решение находится приравниванием нулю частных производных по элементам матрицы W . Равенство (6) можно переписать в следующей форме

$$\frac{\partial}{\partial W[i|p]} \sum_k (M[i|k] - \sum_j W[i|j]X[j|k])^2 = 0$$

Отсюда следует

$$\sum_k M[i|k]X[p|k] = \sum_k \sum_j W[i|j]X[j|k]X[p|k]$$

или

$$(MX^T)[i|p] = (WXX^T)[i|p], \quad \forall(i, p)$$

Это означает, что $MX^T = WXX^T$, а искомая матрица

$$W = (MX^T)(XX^T)^{-1}$$

При этом, возможно, обратную матрицу надо заменить на псевдообратную. Классификация произвольного вектора γ осуществляется следующим образом. Вычисляем

$\mu = W\gamma$ и относим вектор к классу, номер которого совпадает с позицией максимальной компоненты вектора μ .

3.2 Метод Фишера для двух классов

Пусть исходное тренировочное множество векторов длины n разбито на два класса C_1, C_2 $\alpha_i \in C_1, i = 1, \dots, K, \beta_j \in C_2, j = 1, \dots, L$. Выберем вектор γ той же длины, получим числа $x_i = \gamma^T \alpha_i, y_j = \gamma^T \beta_j$. Для полученных чисел найдем средние значения

$$m_1 = \frac{1}{K} \sum x_i, \quad m_2 = \frac{1}{L} \sum y_j$$

и подберем вектор γ таким образом, чтобы

$$J(\gamma) = \frac{(m_1 - m_2)^2}{\sum (x_i - m_1)^2 + \sum (y_j - m_2)^2} \rightarrow \max \quad (7)$$

По определению

$$m_1 = \frac{1}{K} \gamma^T \sum \alpha_i = \gamma^T \alpha,$$

где α некоторый вектор. Аналогично

$$m_2 = \gamma^T \beta,$$

Теперь $m_1 - m_2 = \gamma^T (\alpha - \beta) = \gamma^T \delta$, поэтому

$$(m_1 - m_2)^2 = \gamma^T \delta \delta^T \gamma = \gamma^T A \gamma, \quad (8)$$

где A – матрица. Проводя аналогичные вычисления, получим, что условие (7) принимает вид

$$J(\gamma) = \frac{\gamma^T A \gamma}{\gamma^T B \gamma} \rightarrow \max \quad (9)$$

Можем считать знаменатель в (9) равным 1, после чего задача превращается в обычную задачу оптимизации с ограничениями, решаемую методом Лагранжа. Она сводится к условию параллельности векторов $A\gamma$ и $B\gamma$. По построению, матрица A имеет ранг 1, поэтому вектор $A\gamma$ всегда параллелен вектору δ . Отсюда следует, что вектор γ можно взять в виде $B^{-1}\delta$.

Если теперь вектор γ найден, то классификация осуществляется следующим образом. Для произвольного вектора μ подсчитывают $\gamma^T \mu$, и относят вектор к тому классу, для которого разность $|c - m_i|$ имеет меньшее значение. Более точное выражение для дискриминантной функции можно получить, сделав дополнительные предположения о распределении векторов.

Замечание Условие параллельности векторов $A\gamma$ и $B\gamma$ приводит к множеству решений, но все, кроме отмеченного, при условии $|B| \neq 0$ будут нулевыми.

3.3 Метод Фишера для произвольного числа классов

Для произвольного числа классов существует аналогичная модель, однако, она является более сложной с вычислительной точки зрения. Вместо этого можно использовать метод голосования, представленный выше. Пусть надо решить, принадлежит ли данный вектор α заданному классу C_1 . Считается, что уже построены критерии, различающие класс C_1 со всеми остальными классами. Если для любых двух классов C_1 и $C_i, i > 1$ принимается решение о принадлежности $\alpha \in C_1$, то принимается окончательное решение о принадлежности вектора данному классу. В противном случае отказываются от принятия решения, либо принимается решение о принадлежности указанному классу с некоторым весом меньше 1.

4 Классификация на основе метода Байеса

Наиболее полное изложение этого материала можно найти в [1]

4.1 Классический подход

Предположим, что известны априорные вероятности появления того или иного вектора. Рассмотрим задачу об отнесения вектора к одному из двух классов. Имеется тренировочные множества C_1, C_2 , состоящие из N_1 и N_2 векторов соответственно, но теперь допускается, что множества имеют непустое пересечение. Это означает, что один и тот же вектор α может попасть в каждое из этих множеств с вероятностями $P(\alpha|C_1), P(\alpha|C_2)$. Положим $N = N_1 + N_2, P(C_1) = N_1/N, P(C_2) = N_2/N$. Согласно известной формуле Байеса

$$P(C_1|\alpha) = \frac{P(\alpha|C_1)P(C_1)}{P(\alpha|C_1) + P(\alpha|C_2)P(C_2)}$$

Обозначим через

$$a = \ln \left(\frac{P(\alpha|C_1)P(C_1)}{P(\alpha|C_2)P(C_2)} \right)$$

Тогда

$$P(C_1|\alpha) = s(a) = \frac{1}{1 + \exp(-a)} \quad (10)$$

Функция $s(a)$ носит название логистической сигмоидной (похожей на букву S) функции. Очевидно равенство

$$a = \ln \left(\frac{s}{1-s} \right)$$

Заданный вектор α относится к первому или второму классу в зависимости от того, какая из вероятностей $P(C_1|\alpha), P(C_2|\alpha)$ больше. Данный метод непосредственно распространяется на произвольное число классов. Положим

$$a_k = \ln(P(\alpha|C_k)P(C_k))$$

Тогда

$$P(C_k|\alpha) = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \quad (11)$$

Объект относится к классу, для которого условная вероятность, подсчитанная согласно (11) будет максимальной.

5 Нейронные сети

Рассмотренные дискриминантные функции, основанные на линейных функциях, несут очевидные ограничения. Стандартная feed-forward нейронная сеть представляет новый обширный класс функций. Ни более полное изложение теории нейронных сетей имеется в [2]. Нейронные сети это функции, определенные на векторах $\alpha = \langle a_1, \dots, a_n \rangle$ длины n . Введем обозначение

$$y_k = \omega_k^T \alpha + w_{k,0} \quad (12)$$

Компоненты вектора ω_k называются весами, а числа $w_{k,0}$ – сдвигами. Пусть $h(x)$ некоторая нелинейная функция от одного аргумента. Обозначим через $z_k = h(y_k)$. Набор функций z_k образуют слой нейронной сети, на вход которого подаются входные сигналы α . Выходы слоя сами в свою очередь могут быть входом следующего слоя. Таких слоев может быть много. Наконец, имеется слой, выходы которого являются выходами всей сети. Этот слой называется выходным, а все остальные слои называются скрытыми. Функция $h(x)$ называются функцией отклика нейрона. В качестве функций h обычно выбираются сигмоидные функции, например $h(x) = \tanh(x)$ или $h(x) = 1/(1+\exp(-x))$. Областью значений первой функции является интервал $(-1,1)$, а во втором случае – интервал $(0,1)$. Однако, эти функции не обязаны быть непрерывными. Часто используется $h(x) = \text{sign}(x)$, принимающая только три значения – $-1, 0$ и 1 . Например, сеть с двумя скрытыми слоями и одним выходом представляется в виде

$$z = h_1\left(\sum_{k=1}^M u_k h(y_k) + u_0\right), \quad (13)$$

где y_k определены согласно (12). Обычно, функции отклика внутри одного слоя одни и те же, хотя это не является необходимым условием. В качестве функции отклика можно брать тождественную функцию, это моделирует передачу сигнала с нижнего слоя на выход, минуя промежуточный слой. Подбор весов и сдвигов называют обучением сети.

5.1 Тренировка нейронной сети

Задача заключается в аппроксимации значений многомерной функции, заданной на некотором подмножестве, с помощью нейронной сети. Имеются векторы $\alpha_1, \dots, \alpha_N$ и

числа z_1, \dots, z_N . Требуется построить такую сеть $net(\omega, \alpha)$, чтобы значения $net(\omega, \alpha_i)$ были близки величинам z_i . В качестве критерия близости выберем условие

$$F(\omega) = \sum_i ||net(\omega, \alpha_i) - z_i||^2 \rightarrow \min_{\omega}$$

Выбор указанного критерия удобен тем, что целевая функция становится гладкой, и для отыскания оптимальных параметров можно использовать дифференцирование целевой функции. Для этого находим точки, в которых обнуляется полный дифференциал $dF(\omega) = 0$. Такие точки называются стационарными. Они ищутся, как правило, с помощью итерационных процедур и зависят от выбора начального значения. Наилучшее значение находят экспериментально без обоснования достижения глобального экстремума. Рассмотрим более подробно процедуру отыскания стационарных точек.

Пусть имеется гладкая функция $G(x_1, \dots, x_n)$. В результате замены x_i на $x_i + dx_i$ функция получает приращение

$$dG(x_1, \dots, x_n) = \sum_i \frac{\partial G}{\partial x_i} dx_i$$

Если нужно уменьшить значение функции, то нужно выбрать смещение аргументов в виде

$$\langle dx_1, \dots, dx_n \rangle = -\eta \left\langle \frac{\partial G}{\partial x_1}, \dots, \frac{\partial G}{\partial x_n} \right\rangle, \quad (14)$$

где $\eta > 0$ некоторый коэффициент. Выбор значений этого коэффициента является ключевым моментом. При малом значении будет медленное продвижение к стационарной точке, при большом значении возможен пропуск этой точки. Однако, в случае нейронной сети возникает дополнительная трудность — подсчет частных производных. Метод решения этой проблемы, известный как алгоритм обратного распространения ошибки (back propagation), рассмотрим на примере сети с двумя скрытыми слоями. Пусть

$$y(\alpha_p) = net(\omega, \alpha_p) = \sum_i^K u_i h\left(\sum_j^n v_{ij} a_{pj}\right), \quad \alpha_p = \langle a_{p1}, \dots, a_{pn} \rangle,$$

а функция отклика $h(x) = \tanh(x)$. Положим

$$\beta_i^T = \langle v_{i1}, \dots, v_{in} \rangle, \quad h_{ip} = h(\beta_i^T \alpha_p)$$

Для заданного множества векторов $\alpha_1, \dots, \alpha_N$ требуется подобрать веса таким образом, чтобы

$$F(\omega) = \frac{1}{2} \sum_p^N (y(\alpha_p) - z_p)^2 \rightarrow \min \quad (15)$$

Имеем

$$\frac{\partial F}{\partial u_i} = \sum_p (y(\alpha_p) - z_p) h_{ip} = \sum_p F_p h_{ip}, \quad F_p = y(\alpha_p) - z_p$$

Далее, поскольку $h'(x) = 1 - h^2(x)$,

$$\frac{\partial h_{ip}}{\partial v_{ij}} = (1 - h_{ip}^2) a_{pj},$$

поэтому

$$\frac{\partial F}{\partial v_{ij}} = \sum_p F_p u_i (1 - h_{ip}^2) a_{pj}.$$

Таким образом последовательно вычисляются все частные производные целевой функции по параметрам.

5.2 Практическое вычисление весов

При практической реализации процедуры оптимизации согласно (15) основная трудность заключается в выборе параметра η в (14) и начального приближения для весов. Первый параметр отвечает за скорость сходимости, второй — за положение локального минимума, к которому сходится процедура. Обычно параметр η меняется от шага к шагу, известны многочисленные способы его модификации, которые реализованы в различных пакетах. Что касается выбора начальных значений, то здесь оптимальный выбор делается на основе вычислительных экспериментов, когда начальные значения выбираются с помощью генераторов случайных чисел.

5.3 Метод деформируемого многогранника в задаче тренировки нейронной сети

Метод обратного распространения ошибки годится лишь для дифференцируемых функций. Если это условие не выполнено, либо вычисление производных от функции отклика связано с некоторыми трудностями, можно использовать методы, не использующие производные. Вместо этого можно применить методы, связанные с полным или частичным перебором. Одним из таких методов является метод деформируемого многогранника (Метод Нейдлера-Мида). Это эмпирический метод, предназначенный для отыскания минимума функции. В основе метода лежит предположение, что значение функции в неизвестной точке можно предсказать по значениям этой функции в некоторых известных точках. У этого метода имеется много модификаций, различающихся деталями, но все они имеют общую часть. Пусть имеется функция $f(x_0, x_2, \dots, x_n)$, минимум которой надо найти.

1. Выбираются произвольные $n + 1$ точка $\alpha_0, \alpha_1, \dots, \alpha_n$ и вычисляются значения $z_i = f(\alpha_i), i = 0, \dots, n$.
2. Найденные значения сортируются. Перенумеруем точки, и пусть значения z_0, z_n являются минимальными и максимальными соответственно.
3. Требуется найти новую точку α , значение в которой $z = f(\alpha)$ будет меньше, чем z_n . После этого точка α_n заменяется на α , и процедура повторяется до тех пор, пока не будет найдено приемлемое значение минимума функции

Все модификации метода различаются способом отыскания вектора α . Обычно с этой целью вычисляют вектор

$$\alpha_c = \frac{1}{n} \sum_{i=0}^{n-1} \alpha_i$$

и полагают

$$\alpha = 2\alpha_c - \alpha_n$$

Если окажется, что неравенство $f(\alpha) < f(\alpha_n)$ не имеет места, то пробуют в качестве искомого вектора

$$\alpha_g = \frac{1}{n+1} \sum_{i=0}^n \alpha_i$$

Очевидно, что результат зависит от первоначального выбора точек, и как правило, запускаются несколько процессов с разными начальными данными, после чего выбирают лучшее из найденных значений.

Опишем еще одну модификацию способа отыскания вектора α . В этом случае

1. Вычисляют значения $z_c = f(\alpha_c)$, $z_g = f(\alpha_g)$.
2. Стоят параболу $p(t) = at^2 + bt + c$ по точкам $(t, p(t))$ вида $(0, z_c)$, (l, z_g) , (m, z_n) , где l, m длины векторов $\alpha_c - \alpha_g$, $\alpha_c - \alpha_n$ соответственно.
3. Если $a < 0$, находят корни параболы t_0, t_1 , если $a > 0$, находят точку t_2 , в которой парабола достигает минимума.
4. На прямой, проходящей через точки α_c, α_n откладывают найденные значения t_i , считая точку α_c за начало координат, находят соответствующие векторы, которые проверяют на роль искомого вектора α .

5.4 Нейронная сеть на основе линейной аппроксимации

Рассмотрим задачу классификации, когда тренировочное множество разбито на два класса C_1, C_2 . Нужно построить нейронную сеть, осуществляющую эту классификацию. Задача усложняется, если хотя бы одна из областей имеет сложную конфигурацию (см. Рис.3) Как видно, одна из областей не является выпуклой. Представим ее в виде объединения трех выпуклых областей.

Каждую выпуклую область можно аппроксимировать пересечением областей L_i вида

$$a_i x + b_i y + c_i > 0, \quad i = 1, \dots, N$$

Положим $d_i(x, y) = 0.5 * [\text{sign}(a_i x + b_i y + c_i) + 1]$, считая $\text{sign}(0) = 0$. Перенумеруем области L_i так, что область G_1 задана неравенствами с номерами 1,2,3, область G_2 неравенствами с номерами 4,5,6, область G_3 неравенствами с номерами 7,8,9,10, область G_4 неравенствами с номерами 11, 12, 13. Рассмотрим функции, заданными уравнениями

$$y_1(x, y) = d_1 \wedge d_2 \wedge d_3 \vee d_4 \wedge d_5 \wedge d_6 \vee d_7 \wedge d_8 \wedge d_9 \wedge d_{10}$$

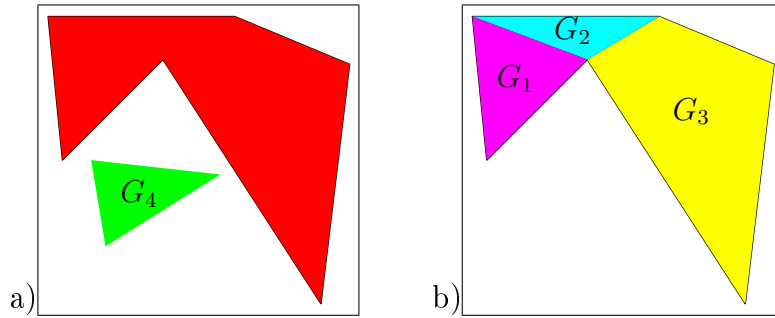


Рис. 3: а) Исходные области. б) Разбиение на выпуклые части

$$y_2(x, y) = d_{11} \wedge d_{12} \wedge d_{13}$$

Точка принадлежит первому множеству, если $y_1(x, y) = 1$, и ко второму множеству, если $y_2(x, y) = 1$. В остальных случаях происходит отказ от принятия решения. Очевидно, можно считать, что функции реализованы нейронной сетью, где функции отклика нейронов являются булевыми функциями.

Данный метод очевидным образом распространяется на любое конечное число классов. Для векторов размерности больше 3 построения менее наглядные, хотя и в этом случае любой полиэдр можно представить в виде объединения выпуклых полиэдров. Однако в этом случае процесс отыскания выпуклых полиэдров потребует значительных вычислений, если число точек велико. Это делается, например, сортировкой точек по первой координате и последующим разбиением на непустые подмножества. Затем внутри каждого из подмножеств производится сортировка по второй координате и т.д. В результате полиэдр будет представлен в виде объединения гиперкубов. Варьируя число элементов в подмножествах, стараются свести общее число гиперкубов к минимуму.

6 Кластеризация

Под кластеризацией понимается разбиение множества векторов на непресекающиеся подмножества, причем в каждое подмножество объединяются векторы, имеющие некоторое общее свойство. Наиболее распространенным свойством является близость к некоторому одному вектору — центру кластеризации. Методы различаются способом нахождения центра и выбранной метрикой для вычисления близости. Следует отметить, что большинство известных алгоритмов носят эмпирический характер, а полезность алгоритма доказывается демонстрацией примеров. Различают алгоритмы, в которых число кластеров задает пользователь, и алгоритмы, автоматически определяющие оптимальное число кластеров.

6.1 Алгоритм K-means

Опишем алгоритм кластеризации, получивший наибольшее распространение и носящий название K-means (K-средние). В этом случае пользователь заранее определяет число K желаемых кластеров, а алгоритм позволяет создать эти кластеры. Исходными данными являются функция Φ для подсчета качества кластеризации, способ пересчета центров кластеризации, функция r для вычисления расстояния между векторами, векторы $\alpha_1, \dots, \alpha_N$, имеющие длину n , число K и число M – предельное возможное число итераций. Общая схема алгоритма выглядит следующим образом:

1. Произвольно генерируются K векторов β_1, \dots, β_K длины n , которые выбираются в качестве начальных центров кластеризации
2. Подсчитываются числа $d_{ij} = r(\beta_i, \beta_j), i \neq j$ и находится $d = \min(d_{ij})$.
3. Образуются K кластеров C_i согласно правилу $r(\alpha, \beta_i) < d/2 \Rightarrow \alpha \in C_i$.
4. Если остались точки, непопавшие ни в какой кластер, то каждая из таких точек α попадает в кластер C_i , для которого $r(\alpha, \beta_i)$ достигает минимума.
5. Подсчитывается значение функции Φ . Если достигнуто желаемое качество кластеризации или исчерпано число возможных итераций, то процесс останавливается. В противном случае пересчитываются векторы β_i , и управление передается на шаг 2.

Приведем примеры компонентов алгоритма. В качестве функции

$$r(\alpha, \beta) = r(\langle x_1, \dots, x_n \rangle, \langle y_1, \dots, y_n \rangle)$$

выбирают евклидово расстояние или его квадрат, $\sum_i |x_i - y_i|$, или $\max_i |x_i - y_i|$. Пересчет центров кластеризации по результатам текущей кластеризации осуществляется согласно формулам

$$\beta_i = \frac{1}{N_i} \sum \alpha_{i'}, \quad \alpha_{i'} \in C_i \quad (16)$$

или $\beta_i = \mathbf{median}(\alpha_{i'},)$ где медиана подсчитывается по каждой координате. В формуле (16) N_i число векторов в кластере C_i . В качестве функции качества кластеризации обычно берется

$$\Phi = \sum_i \frac{1}{N_i} \sum_{i'} r(\alpha_{i'}, \beta_i) \quad \alpha_{i'} \in C_i$$

Вычисление нового центра кластеризации с помощью среднего приводит к чувствительности к отдельным выбросам, от чего свободен метод на основе медианы. Для расстояния в виде квадрата евклидова расстояния известно, что выбор центра согласно (16) приводит к минимизации величины

$$\sum_{i'} r(\alpha_{i'}, \beta_i), \quad \alpha_{i'} \in C_i$$

Выбор числа K является ключевым моментом в данном методе. У пользователя должна быть априорная информация об этом значении. Неудачный выбор числа K приводит к зависимости результат от начального выбора векторов β_i .

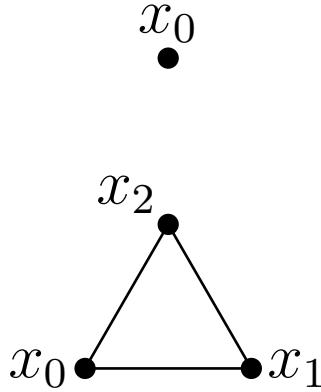


Рис. 4: Пример радиальной интерполяции

7 Радиальные сети

Изложение данного раздела основано на [3]

7.1 Постановка задачи

Рассмотрим классическую задачу интерполяции, когда аргументами являются вещественные векторы длины n , а значения функции — вещественные числа. Требуется найти функцию $h(x)$, удовлетворяющую условиям

$$h(x_i) = t_i, i = 1, \dots, N, x_i \in \mathbf{R}^n, t_i \in \mathbf{R}$$

Решение ищется в форме

$$h(x) = \sum_i v_i f(|x - x_i|) \quad (17)$$

Здесь $|x - x_i|$ обычное евклидово расстояние между векторами, а $f(u)$ — функция, заданная на вещественной оси. Обозначим через A матрицу порядка N , определенную равенствами

$$A[i|j] = f(|x_i - x_j|)$$

Если функция $f(u)$ выбрана таким образом, что $|A| \neq 0$, то коэффициенты v_i в (17) находятся однозначно. Функций, обладающих указанным свойством достаточно много. Проблема заключается в том, что при неудачном выборе этой функции величина $h(x_0)$, когда значение разности $|x_0 - x_i|$ мало, может значительно отличаться от $h(x_i)$. Рассмотрим пример функции $f(u) = u((u-1)+1)$. По определению $f(0) = 0$, $f(1) = 1$. Предположим, что три точки x_1, x_2, x_3 образуют вершины правильного треугольника со стороной 1. В этом случае

$$h(x) = f(|x - x_1|) + f(|x - x_2|) + f(|x - x_3|)$$

является решением задачи $h(x_i) = 2$. Построим точку x_0 . Опустим высоту из вершины x_1 , пометим точку x_0 на продолжении высоты на расстоянии 1 от вершины x_1 (см. Рис.4) . Нетрудно видеть, что $|x_0 - x_3|^2 = 2 + \sqrt{3}$. Отсюда следует, что

$$h(x_0) = f(1) + 2f(\sqrt{2 + \sqrt{3}}) \simeq 8.5$$

То есть, полученное значение далеко от ожидаемого.

7.2 Реализация на основе локализованного базиса

Из представленного выше примера следует, что для получения результата, имеющего практическое значение, используемая функция f не может быть произвольной. Идея, представленная ниже, заключается в том, чтобы большие значения функции были сосредоточены в окрестности нуля. В этом случае, значения $h(x)$ будут близки к $h(x_0)$, если разность $|x - x_0|$ достаточно мала. Функции, обладающие данным свойством, называются локализованными. Примерами таких функций являются

$$f(x) = \exp\left(-\frac{x^2}{a^2}\right),$$

которая получила название "Гауссиан", и функция

$$f(x) = (x^2 + a^2)^{-b^2}$$

Здесь a, b являются параметрами, определяющими степень локализации. При практическом использовании данного метода, когда точек много, требуется выбрать лишь несколько из них для реализации функции. Обычно в качестве таких точек используются центры кластеризации.

8 Методы понижения размерности

Исходная информация для дальнейших исследований представлена, обычно, в виде набора векторов $\alpha_1, \dots, \alpha_N$ одинаковой длины n . Для того, чтобы упростить дальнейшие преобразования полезно попытаться понизить размерность этих векторов, не теряя содержащейся в них информации. Указанные процедуры носят название "методы понижения размерности", или "методами сжатия". Все они организованы следующим образом. Выбираются векторы β_1, \dots, β_m той же размерности, и каждый вектор α_i приближается линейной комбинацией векторов β_j

$$\alpha_i \approx \sum_{k=0}^{m-1} b_{ik} \beta_k \quad (18)$$

В результате исходный вектор α_i длины n заменяется вектором $\langle b_{i1}, \dots, b_{im} \rangle$ длины m . Таким образом, задача сжатия сводится к выбору векторов β_j и к реализации процедуры аппроксимации (18) Здесь мы дадим очень краткий обзор существующих методов сжатия

8.1 Разложение по ортонормированному базису

Наиболее просто рассмотренная выше схема реализуется, когда в пространстве векторов длины n существует ортонормированный базис $\epsilon_1, \dots, \epsilon_n$. В этом случае в качестве векторов β берется некоторое подмножество из этого базиса. При произвольном выборе этого подмножества придется вместе с каждым коэффициентом разложения b_{ik} указывать и номер соответствующего базисного вектора. В силу этого, часто в качестве векторов β выбирают m первых базисных векторов в некоторой нумерации, которая будет одной для всех векторов.

8.1.1 Дискретное преобразование Фурье (ДФТ)

В этом случае базис состоит из векторов

$$\beta_k = \langle \exp(2\pi i k p / n), p = 0, \dots, n - 1 \rangle$$

Если $\alpha = \langle a_0, \dots, a_{n-1} \rangle$, то коэффициенты разложения находятся согласно формулам

$$b_k = \frac{1}{\sqrt{n}} \sum_{p=0}^{n-1} a_p \exp(-2\pi i k p / n),$$

а исходный вектор восстанавливается по коэффициентам разложения

$$\alpha = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} b_k \beta_k$$

Оставив в этой формуле лишь несколько слагаемых, получаем искомое сжатие. Этот базис подходит к ситуации, когда исходные векторы имеют периодическую природу. Номера k трактуются как частоты. Выбираются те векторы, которые охватывают частотный диапазон, наиболее интересный с информативной точки зрения. Основным достоинством преобразования Фурье является то, что модули коэффициентов не чувствительны к циклическим сдвигам компонентов исходных векторов. Недостаток метода заключается в том, что для вещественных векторов коэффициенты разложения оказываются комплексными. В этом случае $b_k = \bar{b}_{n-k}$, поэтому рассматривают лишь коэффициенты с номерами $k \leq [n/2]$.

8.1.2 Дискретное преобразование Хартли (ДНТ)

При изложении данного пункта следуем работе [4]. В качестве базисных векторов выбираются функции

$$\kappa_k = \langle \mathbf{cas}(2\pi i k p / n), p = 0, \dots, n - 1 \rangle, \quad \mathbf{cas}(t) = \cos(t) + \sin(t)$$

Прямая и обратная формулы имеют вид

$$h_k = \frac{1}{\sqrt{n}} \sum_{p=0}^{n-1} a_p \mathbf{cas}(2\pi i k p / n),$$

$$\alpha = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} h_k \kappa_k$$

Поскольку для дискретного преобразования Фурье имеются быстрые схемы вычисления, преобразование Хартли вычисляются с помощью преобразования Фурье согласно формуле

$$h_k = \operatorname{Re}(b_k) - \operatorname{Im}(b_k)$$

Как и в случае преобразования Фурье, результатам преобразования Хартли можно придать частотный характер. Достоинством преобразования Хартли является то, что вещественному вектору будет отвечать вещественный вектор коэффициентов разложения. К недостаткам преобразования следует отнести чувствительность результата к циклическому сдвигу.

8.1.3 Дискретное косинус преобразование (DCT)

Дискретное косинус преобразование оказалось наиболее востребованным способом сжатия вещественных векторов. Оно лежит в основе таких форматов как JPEG и MP3. Изложение ведется согласно [5] В основе DCT лежит DFT. По исходной последовательности a_0, \dots, a_{n-1} строится последовательность

$$\langle z_0, \dots, z_{2n-1} \rangle = \langle a_{n-1}, \dots, a_0, a_0, \dots, a_{n-1} \rangle$$

Вычисляется обычное DFT от последовательности $\langle z_0, \dots, z_{2n-1} \rangle$. В силу симметрии исходных данных дело сводится только к n коэффициентам. В результате получаем

$$v_k = c_k \sum_{p=0}^{n-1} a_p \cos\left(\frac{\pi(2p+1)k}{2n}\right), c_0 = \frac{1}{\sqrt{n}}, c_k = \sqrt{\frac{2}{n}}, k > 0.$$

Обратное преобразование вычисляется по формулам

$$a_p = \sum_{k=0}^{n-1} c_k v_k \cos\left(\frac{\pi(2p+1)k}{2n}\right)$$

8.1.4 Преобразование Адамара (Hadamard transform)

Под матрицей Адамара понимается квадратная матрица составленная из ± 1 , причем разные строки матриц попарно ортогональны. В основе этого преобразования лежит известная серия матриц Адамара порядка $n = 2^q$. Серия строится согласно формуле

$$H_0 = (1), \quad H_{q+1} = \begin{pmatrix} H_q & H_q \\ H_q & -H_q \end{pmatrix}$$

Преобразование задается формулой

$$\langle v_0, \dots, v_{n-1} \rangle^T = H_q \langle a_0, \dots, a_{n-1} \rangle^T$$

Обратное преобразование задается той же формулой

$$\langle a_0, \dots, a_{n-1} \rangle^T = H_q \langle v_0, \dots, v_{n-1} \rangle^T$$

Формально, длина вектора должна быть степенью 2, но заполнив недостающие позиции нулями, можем применить эту технику для векторов произвольной длины. Преобразование Адамара ориентировано на случай, когда векторы имеют целые координаты. Имеются схемы быстрого преобразования, при которых минимизируется число арифметических операций.

8.1.5 Базис Хотелинга-Карунена-Лозва

Этот базис был независимо открыт несколькими авторами в разных контекстах. Рассмотрим простейший из них, принадлежащий Хотелингу (Hotelling). В предыдущих пунктах было изучено разложение по некоторым фиксированным базисам, не зависящих от рассматриваемой задачи. В данном пункте показывается, как можно получить оптимальный базис, основанный на входных данных. Проблема формулируется следующим образом. Даны векторы $\alpha_0, \dots, \alpha_{N-1}$ размерности n . Требуется найти такое подпространство L , $\dim(L) = m < n$, состоящее из векторов той же размерности n и обладающее свойством:

$$\sum_{k=0}^{N-1} |\alpha_k - \alpha'_k|^2 \rightarrow \min, \quad (19)$$

где α' есть проекция вектора α на подпространство L . Выберем в пространстве L ортонормированный базис β_k , $k = 0, \dots, m - 1$. Смысл процедуры сжатия заключается в следующем. Вектор

$$\alpha'_k = \sum_j b_{kj} \beta_j$$

раскладывается по базису, и для его хранения достаточно только m чисел. После замены вектора α его проекцией α' происходит замена вектора длины n на вектор длины m . Решение проблемы (19) сводится к отысканию ортонормированного базиса β_k . Если базис найден, то

$$\alpha'_k = \sum_j (\alpha_j, \beta_j) \beta_j$$

Условие (19) сводится к

$$\sum_k \left| \alpha_k - \sum_j (\alpha_j, \beta_j) \beta_j \right|^2 \rightarrow \min$$

Заменяя квадрат длины вектора скалярным квадратом, сведем дело к условию

$$\sum_{kj} (\alpha_k, \beta_j)^2 \rightarrow \max$$

Считаем, что все векторы являются строками, поэтому

$$(\alpha, \beta) = \alpha\beta^T = \beta\alpha^T, \quad (\alpha, \beta)^2 = \beta\alpha^T\alpha\beta^T$$

Имеем

$$\sum_{kj} (\alpha_k, \beta_j)^2 = \sum_j \beta_j (\sum_k \alpha_k^T \alpha_k) \beta_j^T = \sum_j \beta_j A \beta_j^T, \quad A = \sum_k \alpha_k^T \alpha_k$$

Таким образом, задача свелась к отысканию ортонормированных векторов β_k , для которых

$$\sum_j \beta_j A \beta_j^T \rightarrow \max \quad (20)$$

Это хорошо известная оптимизационная задача. Обозначим через $c_0 \geq c_1 \geq \dots \geq c_{m-1}$ m наибольших собственных значений матрицы A . Пространство L есть подпространство, базисом которого являются собственные векторы, отвечающие собственным числам c_0, \dots, c_{m-1} . В этом пространстве можно выбрать произвольный базис, а максимальное значение суммы в левой части (20) от базиса не зависит. В то же время, базис из собственных векторов обладает рядом замечательных свойств, которые формулируются в вероятностных терминах. Недостатком данного метода является то, что вместе со сжатыми векторами надо хранить и сами базисные векторы, если требуется восстановить исходные векторы (с некоторой погрешностью) по сжатым векторам. В то же время, данный подход нашел большое применение в задачах распознавания образов.

8.2 Аппроксимация с помощью произвольного множества векторов

Изложение ведется согласно [6]. Пусть имеются вектор α и система векторов $\beta_0, \dots, \beta_{N-1}$ из одного и того же евклидова пространства. Единственным условием, наложенным на векторы β_i , является условие нормированности этих векторов. Алгоритм аппроксимации вектора α с помощью векторов β_i выглядит следующий вид

1. Выбирается натуральное число M и число ϵ
2. Полагаем $k = 1, \alpha_k = \alpha$
3. Вычисляются числа $a_j = (\alpha, \beta_j)$. Ищется индекс j_k , для которого $|a_j|$ достигает максимума
4. $\alpha_{k+1} \leftarrow \alpha_k - a_{j_k} \beta_{j_k}$
5. $k \leftarrow k + 1$
6. Если выполнено неравенство $k > M$ или $|\alpha_{k+1}| < \epsilon$, выходим из процедуру, в противном случае переходим к шагу с номером 3

В результате работы процедуры получаем вектор

$$\sum_k a_{jk} \beta_{jk},$$

аппроксимирующий исходный вектор.

Если векторы β_j образуют базу пространства, то в результате работы алгоритма можно добиться любой точности аппроксимации. Действительно, допустим, что имеется базис $\beta_0, \dots, \beta_{n-1}$, и

$$(\forall k) |\alpha_k| > \delta \quad (21)$$

То, что длина вектора α_k не меняется, означает, что $(\forall j)(\alpha_k, \beta_j) \rightarrow 0, k \rightarrow \infty$. Имеем

$$\alpha_k = \sum_i b_i \beta_i,$$

откуда

$$(\alpha_k, \beta_j) = \sum_i b_i (\beta_i, \beta_j) \quad (22)$$

Из независимости векторов β_j следует, что матрица из элементов $c_{ij} = (\beta_i, \beta_j)$ является невырожденной. Из (22) следует, что решая систему относительно b_i получим, что с ростом k эти коэффициенты стремятся к нулю. Это противоречит предположению (21).

С другой стороны, если даже векторы β_j образуют базу пространства, нельзя утверждать, что на некотором шаге получается невязка $\epsilon = 0$. Например, пусть $\alpha(1, 1), \beta_0 = (1, 0), \beta_1 = (1, 2)$. Согласно алгоритму, на каждом шаге невязка получается ортогональной либо вектору β_0 , либо вектору β_1 , поэтому на следующем шаге невязка не может быть нулевой.

8.3 Вейвлет преобразование

Основным источником является [7].

8.3.1 Непрерывное вейвлет преобразование

Пусть имеется функция $f(t)$, определенная на всей вещественной оси. Выбирается функция $\phi(t)$, называемая вейвлетом (wavelet), обладающая свойствами

$$\int_{-\infty}^{\infty} |\phi(t)|^2 dt < \infty, \int_{-\infty}^{\infty} \phi(t) dt = 0$$

Вейвлет преобразованием функции $f(t)$ при помощи вейвлета $\phi(t)$ называется функция

$$W(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \overline{\phi\left(\frac{t-b}{a}\right)} dt, \quad a \neq 0 \quad (23)$$

Существует обратное преобразование, позволяющее восстановить функцию $f(t)$ по $W(a, b)$. При практическом использовании указанного преобразования

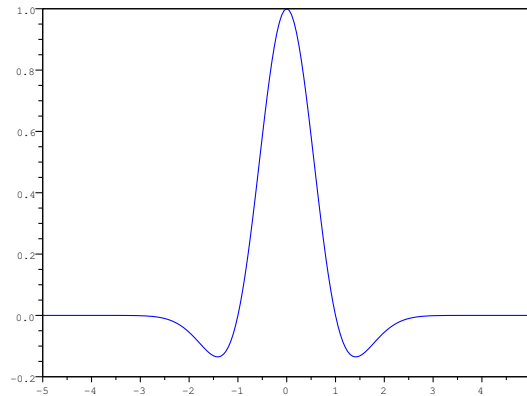


Рис. 5: Пример вейвлета ” мексиканская шляпа”

пользователей имеет дело с цифровым представлением исходной функции. В этой связи, с точки зрения процедур сжатия более интересен подход, представленный в предыдущей секции. В пространстве функций L_2 скалярное произведение задается формулой

$$(f, g) = \int_{-\infty}^{\infty} f(t)\bar{g}(t)dt$$

В этой связи функции вида

$$\phi\left(\frac{t-b}{a}\right)$$

можно рассматривать как множество векторов, с помощью которых аппроксимируется функция $f(t)$. Сначала находятся параметры a_0, b_0 , для которых функция $|W(a, b)|$ достигает максимума. После этого составляется невязка вида

$$f_1(t) = f(t) - \frac{1}{\sqrt{a_0}}W(a_0, b_0)\phi\left(\frac{t-b_0}{a_0}\right),$$

после чего процедура продолжается так, как указано выше. На практике, данный подход оказывается эффективным, когда мощность функции $f(t)$ сосредоточена на малом интервале. Но при этом и функция $\phi(t)$ должна обладать тем же свойством. На Рис.5 представлен пример такой функции, получившей название ”мексиканская шляпа” Эта функция имеет максимум в точке 0. Параметр b_0 дает положение максимума исходной функции $f(t)$, а параметр a_0 – форму кривой.

8.3.2 Дискретное вейвлет преобразование (DWT)

Теория вейвлет преобразования является серьезным достижением в теории обработки сигналов за последние десятилетия. Различают непрерывное и дискретное вейвлет преобразования. О непрерывном преобразовании будет сказано ниже.

Пусть имеется сигнал $\gamma = \langle g_0, \dots, g_n \rangle$. Под мощностью сигнала будем понимать величину $Pow = \sum g_k^2$, а носителем вектора назовем множество индексов с ненулевыми компонентами. Любой сигнал может быть разложен по каждому из рассмотренных выше базисов, однако, все эти базисы обладают одним свойством. Если β произвольный базисный вектор $\beta = \langle b_0, \dots, b_{n-1} \rangle$, то мощность любого интервала $\langle b_k, \dots, b_m \rangle$ одной и той же длины будет примерно одинаковой. Это означает, что с точки зрения компрессии, эффективной процедура сжатия будет для векторов, которые обладают тем же свойством. Если же предыдущее условие для сигнала не имеет места, например, мощность сигнала сосредоточена на коротком интервале, то использование рассмотренных выше базисов приводит к тому, что коэффициенты разложения будут иметь близкие по модулю значения. Для аппроксимации подобных векторов нужен базис, обладающий тем же свойством — мощность (носитель) сосредоточена на небольшом интервале. Рассмотрим класс дискретных вейвлет преобразований, связанных с именем Добеши (I. Daubechies). Ниже во всех суммах индексы переменных пробегают все возможные значения из области определения.

Пусть удалось найти числа p_0, \dots, p_{2N-1} , обладающих свойством

$$\sum_{k \geq 0} p_k p_{k+2m} = 0, \quad \sum_{k=0}^{2N-1} p_k = 2, \quad \sum_{k=0}^{2N-1} p_k^2 = 2 \quad (24)$$

для всех натуральных значений m , для которых (24) имеет смысл. По этим числам построим числа $q_k = (-1)^k p_{2N-k-1}$. Имеем

$$\sum_{k \geq 0} q_k q_{k+2m} = \sum_{k \geq 0} (-1)^k p_{2N-k-1} (-1)^{k+2m} p_{2N-k-2m-1}$$

Если $k' = 2N - 2k - 2m - 1$, то $2N - k - 1 = k' + 2m$, откуда следует, что

$$\sum_{k \geq 0} q_k q_{k+2m} = 0$$

Далее,

$$\sum_{k \geq 0} p_k q_{k+2m} = \sum_{k \geq 0} p_k (-1)^k p_{2N-k-2m-1}, \quad m = 0, 1, \dots$$

Последнее равенство вытекает из того, что число слагаемых в сумме четное и вместе со слагаемым

$$p_k (-1)^k p_{2N-k-2m-1},$$

отвечающим индексу k , в сумме присутствует слагаемое, отвечающее индексу $2N - k - 2m - 1$. Указанные индексы имеют противоположные четности, поэтому соответ-

ствующие слагаемые уничтожатся. Это означает, что матрица

$$Q = \frac{1}{\sqrt{2}} \begin{pmatrix} \cdot & p_0 & p_1 & p_2 & \cdot & \cdot & p_{2N-1} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & p_0 & p_1 & p_2 & \cdot & \cdot & p_{2N-1} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & p_0 & p_1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & q_0 & q_1 & q_2 & \cdot & \cdot & q_{2N-1} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & q_0 & q_1 & q_2 & \cdot & \cdot & q_{2N-1} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & q_0 & q_1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

является ортогональной. Матрица естественным образом разбивается на два блока:

$$Q = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}$$

и имеет бесконечную размерность. Предположим, что имеется вектор $\alpha = \langle a_0, a_1, \dots, a_{2M-1} \rangle$. Этот вектор можно рассматривать как вектор бесконечной длины, в котором все остальные компоненты нулевые. Будем обозначать этот вектор тем же символом. Найдем произведение $\beta^T = Q\alpha^T$. Вектор α восстанавливается по вектору β умножением на матрицу Q^T . Вектор β разбивается на две части: $\beta = \langle \delta, \gamma \rangle$, где $\delta^T = Q_1\alpha^T$, $\gamma^T = Q_2\alpha^T$. Каждая координата вектора γ имеет вид

$$g_m = \sum_{k \geq 0} p_k a_{k+2m},$$

а каждая координата вектора δ имеет вид

$$d_m = \sum_{k \geq 0} q_k a_{k+2m},$$

где m целое число. Введем нумерацию для строк матрицы Q

$$\mu_k = \langle 0, \dots, 0, p_0, \dots, p_{2N-1}, 0, \dots, \rangle$$

$$\nu_k = \langle 0, \dots, 0, q_0, \dots, q_{2N-1}, 0, \dots, \rangle$$

где k любое целое число. Одной из строк типа μ присвоим индекс 0, все строки, расположенные выше, будут иметь отрицательные индексы, а расположенные ниже — положительные. Аналогичные нумерация производится для строк типа ν . В этих обозначениях вектор α есть линейная комбинация векторов μ_k и ν_k .

$$\alpha = \sum_k [(\alpha, \mu_k)\mu_k + (\alpha, \nu_k)\nu_k] \quad (25)$$

Заметим, что коэффициенты в (25) есть элементы вектора β .

Векторы μ_k, ν_k имеют носитель длины не более $2N$, поэтому они ориентированы на аппроксимацию векторов, мощность которых сосредоточена на небольшом отрезке.

Разложение (25) приводит к ряду замечательных следствий. Коэффициенты g_m можно рассматривать как результат фильтрации вектора α FIR фильтром с коэффициентами $c_0, \dots, c_{2N-1}, c_k = p_{-k}$, причем в отфильтрованном сигнале удалены компоненты с нечетными индексами. Аналогично интерпретируются коэффициенты d_m . При этом первый фильтр является фильтром низких частот, а второй фильтр — фильтром высоких частот. Другими словами, векторы γ и δ можно рассматривать как результаты фильтрации исходного вектора α . Это обстоятельство позволяет осуществить процедуру кратномасштабированной (multiresolution) аппроксимации исходного вектора. Суть процедуры заключается в следующем. На первом шаге выделяется низкочастотная часть вектор γ , который обозначим через γ_1 . Затем та же процедура применяется к вектору γ_1 , в результате чего получается вектор γ_2 . Поскольку рассматриваемое преобразование является ортогональным, на каждом шаге сравнивается длина вектора γ_k с длиной исходного вектора α . Как только расхождение превысит заданный порог, процедура останавливается. Выбрав высокочастотную составляющую нулевой, по каждому вектору γ_k можно восстановить с помощью обратного преобразования вектор $\bar{\alpha}$ — полученную аппроксимацию и сравнить ее с оригиналом по любому критерию. Если окажется, что основная мощность сигнала сосредоточена в высокочастотной области, аналогичную процедуру можно применить и к ней. Коэффициенты вейвлетов Добеши просчитаны для различных N . Вид нужного вейвлета выбирается, исходя из качества аппроксимации, достигнутого за данное число шагов. Вид материнской функции можно получить, восстанавливая сигнал, для которого $\gamma_k = \langle 1 \rangle$.

9 Сжатое представление последовательностей произвольной длины

Выше была рассмотрена ситуация, когда требовалось получить сжатое представление векторов одинаковой длины. Представим теперь, что объектом исследования являются звуковые файлы, полученные в результате записи одного и того же слова. Очевидно, что теперь все записи имеют разные длины, и для их сжатого описания должны применяться методы, основанные на усреднении некоторых параметров. Если речь идет о звуковых файлах, то такими параметрами являются разного рода спектральные характеристики, о которых будет сказано при изучении проблемы аутентификации по голосу. В данном пункте рассмотрим подход, применимый к произвольным векторам, хотя в его основе лежит описание скрытых марковских цепей. Пусть задана последовательность $\alpha = \langle a_0, \dots, a_{N-1} \rangle$, элементы которой принадлежат конечному множеству M , $|M| = m$. Элемент этого множества с индексом i обозначим через $M[i]$. Построим матрицу P размера $m \times m$ по следующему правилу. Положим B_i равным числу компонент в α , совпадающих с $M[i]$, а через B_{ij} обозначим число пар вида $\langle a_k, a_{k+1} \rangle$, где $a_k = M[i], a_{k+1} = M[j]$. Теперь

$$P[i|j] = \frac{B_{ij}}{B_i}$$

Данное определение соответствует определению матрицы перехода однородной марковской цепи с m состояниями. Интуитивно ясно, что если две последовательности отвечают одному и тому же явлению, в чередовании элементов этой последовательности должна быть какая-то закономерность. Матрица P и предназначена для описания этой закономерности. На практике в качестве множества M выбирается некое разбиение множества всех возможных значений элементов последовательности. Понятно, что результат будет зависеть от выбора этого разбиения. Обычно, разбиение проводят на основе гистограммы, построенной по множествам последовательностей так, чтобы в каждое подмножество попало примерно одно и то же множество элементов последовательности.

Список литературы

- [1] *C. M. Bishop* Machine Learning. Springer, 2006, 738 p.
- [2] *C. Хайкин* Нейронные сети: полный курс. Вильямс, 2008, 1104 с.
- [3] *C.M. Bishop* Neural Networks for Pattern Recognition. Clarendon Press. Oxford, 1995. 482 p.
- [4] *P. Брейсуэлл* Преобразование Хартли. М.:Мир, 1990. 175 с.
- [5] *A.K. Jain* Fundamentals of digital image processing. Prentice Hall, 1989, 570 p.
- [6] Распознавание образов и медицинская диагностика/ Под редакцией Ю.И.Наймарка, М.:Наука, 1973. 328 с.
- [7] *C. Малла* Вейвлеты в обработке сигналов. М.:Мир, 2005. 671 с.