



СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
SIBERIAN FEDERAL UNIVERSITY



ПРОСПЕКТ СВОБОДНЫЙ- 2017

МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ МОЛОДЫХ УЧЕНЫХ

ЭЛЕКТРОННЫЙ СБОРНИК МАТЕРИАЛОВ
МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ СТУДЕНТОВ,
АСПИРАНТОВ И МОЛОДЫХ УЧЕНЫХ
“ПРОСПЕКТ СВОБОДНЫЙ 2017”
ПОСВЯЩЕННОЙ ГОДУ ЭКОЛОГИИ В РФ

КРАСНОЯРСК, СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

17-21 АПРЕЛЯ 2017 Г.

**Министерство образования и науки Российской Федерации
ФГАОУ ВО «Сибирский федеральный университет»**

Проспект Свободный - 2017

Материалы научной конференции
посвященной Году экологии в Российской Федерации
17-21 апреля 2017 г.

Электронное издание

Красноярск
СФУ
2017 г.

Прикладная математика и информатика

СРАВНЕНИЕ ТЕХНОЛОГИЙ РАСПАРАЛЛЕЛИВАНИЯ ВЫЧИСЛЕНИЙ НА ПРИМЕРЕ РЕШЕНИЯ ЗАДАЧИ О КРУЧЕНИИ СТЕРЖНЯ ПРЯМОУГОЛЬНОГО СЕЧЕНИЯ

Керп А.С.

Научный руководитель канд. физ.-мат. наук Кучунова Е.В.

Сибирский федеральный университет

Применение параллельных вычислительных систем является стратегическим направлением развития вычислительной техники. Это обстоятельство вызвано не только ограничением максимально возможного быстродействия обычных последовательных ЭВМ, но и постоянным наличием вычислительных задач, для решения которых возможностей существующих средств вычислительной техники оказывается недостаточно.

Подобные системы позволяют выполнять необходимые вычисления с меньшими затратами времени, достигая ускорения процесса решения вычислительной задачи.

В качестве тестовой задачи была рассмотрена задача о кручении призматического стержня прямоугольного сечения, определяемая как задача нахождения функции $\varphi = \varphi(x, y)$, удовлетворяющей в области определения $D \subset \mathbb{R}^2$ уравнению:

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = -2, \quad \forall (x, y) \in D \quad (1)$$

с краевым условием на всей границе:

$$\varphi(x, y) = 0, \quad \forall (x, y) \in \partial D, \quad (2)$$

где φ – функция напряжения, $D = \{|x| \leq a; |y| \leq b\}$, $a \geq b$.

С известным решением:

$$\varphi = b^2 - y^2 + \frac{32b^2}{\pi^3} \sum_{s=0}^{\infty} \frac{(-1)^{s+1}}{(2s+1)^3} \frac{ch \frac{(2s+1)\pi x}{2b}}{ch \frac{(2s+1)\pi a}{2b}} \cos \frac{(2s+1)\pi y}{2b}. \quad (3)$$

Для решения поставленной задачи в данной работе применяются методы конечных разностей и конечных элементов.

Метод конечных разностей (МКР) использует метод сеток: область D представляется в виде дискретной сетки узлов $D_h = \{(x_i, y_j): x_i = ih, y_j = jh, 0 \leq i, j \leq N\}$, где величина N – размерность сетки, а h – шаг сетки (рис.1).

Через φ_{ij} обозначается аппроксимация функции $\varphi(x, y)$ в точках (x_i, y_j) .

Для аппроксимации используется метод верхней релаксации:

$$\varphi_{ij}^{(k+1)} = \frac{1}{4} w \left(\varphi_{i,j-1}^{(k+1)} + \varphi_{i,j+1}^{(k)} \right) + \varphi_{i-1,j}^{(k+1)} + \varphi_{i+1,j}^{(k)} - 4 \left(1 - \frac{1}{w} \right) \varphi_{ij}^{(k)} + 2h^2, \quad (4)$$

где k – номер приближения, w – задаваемый параметр ($1 < w < 2$).

Последовательность решений, получаемых методом сеток, равномерно сходится к решению задачи Дирихле, а погрешность решения имеет порядок h^2 .

Для создания параллельных приложений используются специализированные системы поддержки параллельного программирования, такие как интерфейс передачи данных MPI и OpenMP.

При построении параллельных программ для решения задачи возможны два различных способа разделения данных между вычислительными узлами – ленточная схема или блочная схема разбиения вычислительной сетки (рис. 2).

При ленточном разбиении область расчетов делится на горизонтальные или вертикальные полосы, и распределяются между процессорами для обработки. На каждую полосу должны быть продублированы граничные строки предшествующей и следующей полос вычислительной сетки для поддержания итерационного процесса (рис. 3). Пересчет этих строк происходит в полосах своего исходного месторасположения.

В блочной схеме область расчетов делится на блоки. Обмен происходит как граничными строками, так и граничными столбцами (рис. 3).

Организацию параллельных вычислений на системах с распределенной памятью обеспечивает интерфейс передачи данных MPI.

Блокирующие операции обмена данными между вычислительными узлами в MPI приостанавливают выполнение процессов до момента завершения работы вызванных функций. Неблокирующие команды в таких ситуациях позволяют выполнять функции обмена данными без блокировки, совмещая процессы передачи сообщений и вычислений там, где это возможно.

Для вычислительных систем с общей памятью систем используют технологию OpenMP, которая реализует параллельные вычисления с помощью многопоточности.

При работе в общем адресном пространстве появляется проблема взаимной блокировки потоков при доступе к общим данным. В OpenMP это может быть организовано при помощи неделимых операций, механизма критических секций или замков, которые приостанавливают выполнение всех потоков, кроме одного. Для повышения продуктивности может быть использована директива reduction, которая создает локальные копии общих данных для всех потоков, и над этими копиями после выполнения всех операций параллельной области выполняется некоторая заданная операция.

Для оценки параллельных алгоритмов были произведены замеры времени выполнения параллельной части программы, по которым определялись ускорение и эффективность.

Ускорением параллельного алгоритма называют величину $S_p = \frac{T}{T_p}$, где T - время работы последовательной программы, T_p - время работы параллельной программы, с использованием p процессоров.

Эффективностью называют величину $E = \frac{S}{p}$, где p – число используемых процессоров.

При проведении анализа параллельных программ, удалось выяснить, что для данной задачи использование неблокирующих операций передачи данных в MPI значительно повышает ускорение и эффективность программы, при чем наиболее эффективной себя показала программа с ленточной схемой разбиения данных, за счет того, что в блочной схеме количество операций обмена данными, которые замедляют вычислительный процесс, в несколько раз больше. В OpenMP применение директивы reduction дало также существенный прирост к ускорению и эффективности параллельной программы.

На основе наиболее эффективных параллельных программ была построена гибридная программа, совмещающая технологии MPI и OpenMP. За счет MPI реализуется разделение вычислительной задачи между процессорами, а OpenMP распределяет задачу между ядрами

каждого процессора. За счет данного объединения удалось добиться роста эффективности и ускорения параллельной программы (рис. 4).

Следующим этапом работы стало решение поставленной задачи Дирихле методом конечным элементов. В настоящее время метод конечных элементов (МКЭ) является одним из наиболее популярных методов решения краевых задач.

В МКЭ исходная область разбивается конечные элементы (рис. 5), на которых искомая непрерывная функция аппроксимируется кусочно-непрерывной. В итоге МКЭ сводится к решению СЛАУ с сильноразряженной матрицей $A\Phi = b$. Для решения получившейся СЛАУ используется итерационный метод Гаусса-Зейделя. СЛАУ вида $A\Phi = b$ представляется в виде $X = CX + d$. Далее ведется итерационный процесс:

$$\Phi_i^{(k+1)} = \sum_{j=1}^{i-1} c_{ij} \Phi_j^{(k+1)} + \sum_{j=i}^n c_{ij} \Phi_j^{(k)} + d_i, \quad i = 1 \dots N, \quad (5)$$

где i – номер компоненты, k – номер итерации.

Метод равномерно сходится к решению задачи Дирихле, а погрешность решения имеет второй порядок.

При разработке параллельного алгоритма матрица C и вектор d распределяются ленточно между процессорами, обновляя на каждой итерации свою часть вектора Φ . После очередной итерации процессоры обмениваются этими частями.

Для гибридной программы использовались методы MPI с неблокирующими функциями и методы OpenMP с директивой reduction.

Создание гибридной программы может дать выгоду применительно не ко всем задачам. Решая поставленную задачу методом конечных элементов, добиться большей эффективности параллельной программы за счет объединения технологий распараллеливания не удастся, т.к. при решении СЛАУ используется достаточно малое количество коммуникационных операций и небольшой объем передаваемых данных, поэтому OpenMP не дает выгоды для вполне эффективной MPI-программы (рис. 6).

При решении задачи МКР составление гибридной программы имело смысл при работе с задачами больших размерностей, так как OpenMP справляется в этом случае с задачами распараллеливания эффективнее, чем MPI, нагруженная большим числом коммуникационных операций. Однако параллельные программы для МКЭ имеют значительно лучшие показатели ускорения и эффективности, которые практически не изменяются при росте количества используемых вычислительных узлов.(рис. 7).

Список использованных источников

1. Гергель В.П. Теория и практика параллельных вычислений: М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. – 423 с.
2. Лурье А.И. Теория упругости. – М.: Наука, Главная редакция физико-математической литературы, 1970. — 940 с.
3. Трудоношин В.А., Пивоварова Н.В. Системы автоматизированного проектирования: С40 В 9-ти кн. Кн. 4. Математические модели технических объектов: – М.: Высш. шк., 1986. – 160с.

РАЗРАБОТКА ФОРМАТА ХРАНЕНИЯ GPS-ТРЕКОВ

Гончарик К.В.

Научный руководитель кандидат физ.-мат. наук Цыганок Д. А.

Институт математики и фундаментальной информатики СФУ

В связи с распространением мобильных устройств возросла популярность мобильных гео-информационных систем и картографических сервисов и навигаторов в виде приложений: 2ГИС, Яндекс.карты, google.maps, bing maps. А так же различные приложения использующие базу openstreetmap, такие как: osmand, mapsme, 7ways и др. В несжатом виде такие базы могут иметь огромный размер (например база openstreetmap по состоянию на 6.03.17 в зависимости от формата, может занимать от 37 до **700 GB**)

В связи с довольно сильной ограниченностью ресурсов мобильных устройств по размеру дискового хранилища производители приложений создают собственные форматы как для хранения баз ГИС, так и для хранения сгенерированных пользователями данных (gps-треки, POI, заметки и др.)

В данной работе рассматривается метод сжатия GPS-треков - набор измерений приёмника систем глобального позиционирования, упорядоченный по времени. Большинство программ, для сохранения GPS-треков используется формат GPX, это текстовый формат хранения и обмена данными систем глобального позиционирования, основанный на XML — поэтому файлы в данном формате достаточно объёмны. Для решения этой проблемы была проведена работа по исследованию баз треков openstreetmap и gpslib и нахождению нового способа хранения GPS-треков.

Так как основной информацией в gps-треке являются координаты точек (возможно с высотой) и время из записи — эти данные необходимо сохранять. Остальные данные трека несут вспомогательный характер и в целях сохранения дискового пространства ими можно пожертвовать. При разработке формата не рассматривались текстовые, «человеко-читаемые» способы записи, и было выбрано хранение в бинарном виде.

Так же для уменьшения занимаемого размера было использовано:

1. Координаты в fixed-point 9.23;
2. Опционально – закругление координат;
3. Дельта-кодирование;
4. Кодирование хатфмана по части алфавита;
5. Использование формата Varint;

6. Опционально – использование snappy-сжатая.

В результате комбинирования данных подходов, был создан формат хранения 7WT, на рисунке 1 изображён пример GPS-трека: позволяющий сохранять gpx-треки, включая координаты, высоту и время.

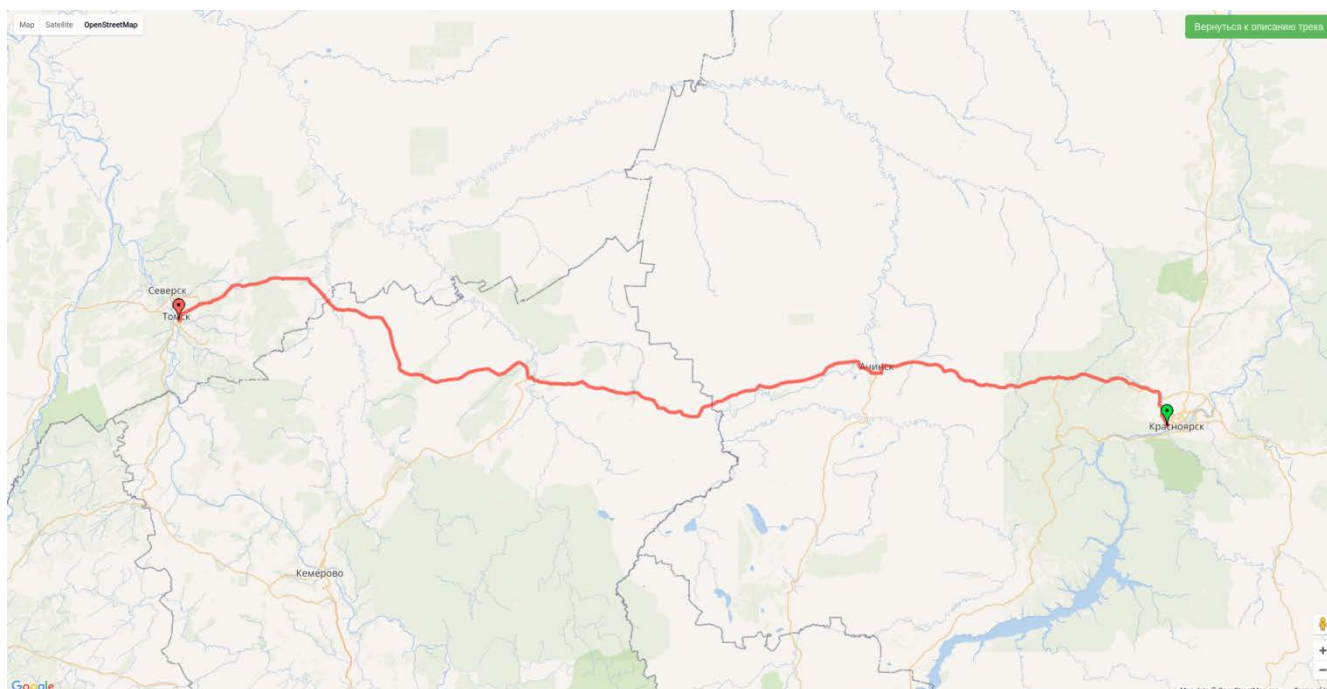


Рис. 1 – GPS-трек gpplib#66727 Красноярск-Томск пользователя vas@mpeks.tomsk.su

Таблица 1 – Сравнение размеров треков в различных форматах.

ИД трека на gpplib	Размер файла байт					
	GPX	PLT	KML	MPS	GDB	7WT
66727 (рис 1.)	3 292 971	2 010 208	910 286	853 080	660 471	57 821
12309	1 222 837	705 021	327 988	316 799	245 299	14 511
1337	850 376	509 975	247 677	220 475	170 723	23 553
265	66 579	52 291	25 772	22 878	11 864	5 974
228	420 774	236 067	111 483	105 401	81 641	30 170

В итоге в результате работы был создан алгоритм хранения GPS-треков использующий минимальное, среди популярных форматов, дисковое пространство.

Список использованных источников

1. Ватолин, Д. Все о сжатии данных [Электронный ресурс] / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин, Е. Шелвин, Д. Шкарин // Лаборатория компьютерной графики и мультимедиа при факультете вычислительной математики и кибернетики (ВМиК) МГУ. 2001-2011. Режим доступа: <http://compression.ru/>.
2. GPX: the GPS Exchange Format [Электронный ресурс] / Режим доступа: <http://www.topografix.com/gpx.asp>.
3. MAPS.ME — Offline OpenStreetMap maps for iOS and Android [Электронный ресурс] / Исходные коды проекта maps.me // Режим доступа: <https://github.com/mapsme/omim>.
4. google/snappy a fast compressor/decompressor [Электронный ресурс] / Исходные коды проекта google/snappy // Режим доступа: <https://github.com/google/snappy>.

ИССЛЕДОВАНИЕ МЕТОДОВ РЕШЕНИЯ ЗАДАЧИ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ

Докучаев Д.А.,

Научный руководитель канд. физ.-мат. наук, доцент Баранова И.В.

Сибирский Федеральный Университет,

Институт математики и фундаментальной информатики

Задача распознавания изображений относится к широкому классу задач распознавания образов. Распознавание образов – раздел статистического анализа данных, объединяющий методы разбиения объектов по нескольким категориям или классам. Объекты называются **образами**.

Будем считать, что все объекты разбиты на конечное число классов. Для каждого класса известно и изучено конечное число объектов – прецедентов. **Прецедент** – это образ, правильная классификация которого заранее известна. **Задача распознавания образов** состоит в том, чтобы отнести новый распознаваемый объект к какому-либо классу.

Распознавание образов является востребованным и успешно развивающимся направлением в современной теоретической информатике и анализе данных. Его методы имеют широкий спектр применений практически во всех областях человеческой деятельности, связанных с изучением объектов и процессов: медицине, биологии, химии, маркетинге, психологии, социологии, менеджменте и других.

Целью данной работы является исследование постановки задачи распознавания образов и основных методов распознавания, а также особенностей их работы в ситуации распознавания изображений.

Задача классификации и распознавания образов

Формулировка задачи распознавания образов выглядит следующим образом: имеется множество объектов $X = \{x_1, \dots, x_n\}$, где $x: x \in X$ – объект распознавания. Тогда $g(x): X \rightarrow M, M = \{1, \dots, m\}$ – индикаторная функция, разбивающая пространство образов X на m непересекающихся классов. Индикаторная функция неизвестна.

Пусть F – пространство признаков, функция $f(x): F \rightarrow X$ ставит в соответствие каждому объекту $x_i \in X$ точку $f(x)$. Вектор $f(x)$ называется **образом объекта** $x \in X$. В пространстве образов признаков определены непересекающиеся множества точек $K_i \in F, i = \overline{1, m}$, соответствующие образам одного класса.

Функция $\bar{g}(f): F \rightarrow M$ называется решающим правилом, оно представляет собой оценку для $g(x)$ на основании $f(x)$, то есть $\bar{g}(f) = \bar{g}(f(x))$.

Пусть $f_j = f(x_j)$, $j = \overline{1, N}$ – доступная информация о функциях $f(x)$ и $g(x)$, но сами эти функции неизвестны. Тогда (g_j, f_j) – есть множество прецедентов.

Требуется построить такое решающее правило $\bar{g}(f)$, чтобы распознавание и классификация проводилось с минимальным числом ошибок.

Как уже было сказано выше, каждый объект описывается набором своих характеристик, называемых *признаками*. Признаки могут быть следующих типов: *бинарный* признак (0 или 1), *номинальный (качественный)* признак, *порядковый* признак или *количественный* признак. Самой распространенной ситуацией является кластеризация объектов, у которых все признаки являются количественными, т.е. когда $F = \{F_1, \dots, F_n\} \in \mathbb{R}^n$.

Для нашей задачи распознавания изображений в качестве множества объектов берутся изображения.

Практическая задача распознавания и классификации

Рассмотрим задачу распознавания пешеходов на изображении. Существует множество разных подходов для решения данной задачи. Выделяют стохастические, детерминистские и структурные подходы в распознавании изображений. В данной работе будут использоваться исключительно детерминистские подходы, такие как метод опорных векторов, искусственные нейронные сети прямого распространения и метод потенциальной функции. Ниже будет краткое описание перечисленных методов.

Метод опорных векторов является одним из быстрых методов по сходимости и по распознаванию. Существует множество градиентных методов решения условной оптимизационной задачи. Наиболее часто используемые подходы – это метод наискорейшего спуска и метод сопряженных градиентов. Первый является итерационным, второй является конечным и гарантирует сходимость не более чем за N шагов, благодаря тому, что целевая функция представляет собой отрицательно определенный квадратичный функционал. Однако метод имеет серьезный недостаток: так как решение ищется в области $\lambda_i \geq 0$, в некоторых ситуациях задача сводится к перебору на множестве всех подмножеств. Нейронные сети отличаются хорошей сходимостью решений и возможностью аппроксимации нелинейной функции. Существует множество эвристик, которые позволяют ускорять процесс обучения и получать хорошие результаты. Нейросети применяют не только в классификации, но и в регрессии, и в кластеризации.

Метод потенциальных функций обладает медленной сходимостью, но он достаточно прост для реализации. Он позволяет находить решающую функцию, которая классифицирует линейно неразделимые объекты.

Метод опорных векторов SVM

Используя данный метод нужно оценить, является ли задача линейно разделимой или нет. Рассмотрим задачу нахождения наилучшего в некотором смысле линейного разделения множества векторов на два класса. Пусть имеется множество прецедентов (X, Y) , где $X = \{x_1, \dots, x_N\}$ - обучающая выборка, $Y = \{y_1, \dots, y_N\}$ - множество меток двух классов ω_1 и ω_2 соответственно. Требуется по обучающей выборке построить линейную решающую функцию $f(x)$, которая удовлетворяла бы условию $f(x_i) > 0 \forall x_i \in \omega_1, f(x_i) < 0 \forall x_i \in \omega_2$.

$$y_i = \begin{cases} 1, & x_i \in \omega_1 \\ -1, & x_i \in \omega_2 \end{cases} \quad (1)$$

Без ограничения общности можно считать, что метки классов равны (1). Тогда поставленную выше задачу можно переформулировать следующим образом. Требуется найти линейную решающую функцию $f(x)$, которая бы удовлетворяла условию $y_i((w, x_i) + b) \geq 1, i = \overline{1, N}$, где w - вектор весовых коэффициентов, b - некоторое число. Тогда разделяющей два класса гиперплоскостью будет $(w, x) + b = 0$. Нетрудно видеть, что и все гиперплоскости вида $(w, x) + b' = 0$, где $b' \in (b-1, b+1)$ также будут разделяющими. На самих граничных плоскостях может находиться некоторое число (не меньше двух) обучающих векторов. Эти векторы называются **опорными**.

Идея метода состоит в том, чтобы расстояние между плоскостями было как можно больше. Эта проблема решается с помощью минимизации $\|w\|$. Данная задача сводится к нахождению критических точек функции

$$\Phi(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \left\| \sum_{i=1}^N \lambda_i y_i x_i \right\|^2 \quad (2)$$

Искусственная многослойная нейронная сеть

Искусственный нейрон - обобщенная модель человеческого нейрона, который на вход получает некоторое значение и на выход передает обработанное активационной функцией полученное значение. **Синапс** - связь между двумя нейронами, которая обладает некоторым весом.

Многослойной нейронной сетью - сеть, у которой помимо входного и выходного слоя существует хотя бы один скрытый слой. Основным методом обучения такой сети

является алгоритм обратного распространения ошибки. В качестве функции активации зачастую используют сигмоидную функцию либо гиперболический тангенс.

Алгоритм работы нейронной сети по распознаванию выглядит следующим образом:

1. Выполняется инициализация сети: определяется количество слоёв M и количество нейронов на каждом слое m , и они связываются синапсами.
2. Весам синапсов присваиваются случайные значения от 0 до 1.
3. На вход подается значение x .
4. Вычисляются все значения активации нейронов (в обычном режиме функционирования).
5. Начинается корректировка весов синапсов (в обратном направлении).

Корректировка весов сети делается по правилу Видроу — Хоффа. Правило Видроу — Хоффа, называемое также $\square$ - правилом, подразумевает корректировку весов в соответствии с выражением (3)

$$\Delta w_i = \mu \delta x_i \quad (3)$$

Где μ - коэффициент скорости обучения, x_i – значения, поступающие по входным связям.

Этот метод минимизирует ошибку, используя метод градиентного спуска. Таким образом, запуская итерационный процесс и выбирая в качестве критерия остановки минимизацию ошибки на выходном слое, сеть обучается. Для повышения скорости сходимости и во избежание паралича сети применяют различные эвристики в алгоритме обучения.

Метод потенциальной функции

Каждая предъявляемая выборка может быть представлена точкой в пространстве признаков. Представим, что в каждой такой точке помещен электрический заряд. В некоторой произвольной точке M совокупность всех зарядов создает электрический потенциал V , являющийся суммой отдельных потенциалов, создаваемых каждым отдельным зарядом. Известно, что величина потенциала вычисляется по следующей формуле (4):

$$V = \frac{1}{4\pi\epsilon_0} \sum_i \frac{q_i}{r_i} = \sum_i V_i, \quad (4)$$

где q_i - заряд в точке p_i ; r_i - расстояние от точки p_i до точки M . Потенциал выражается функцией, симметричной относительно точки, в которой помещен заряд. В этой точке потенциал, по определению, равен бесконечности. Линии, соединяющие точки равного потенциала, называются эквипотенциальными.

Можно представить, что всякое облако точек, отображающих некоторый класс, выглядит как некое потенциальное плато, отделен от другого плато, отображающего другой класс, глубокой долиной, потенциал которой минимален или даже равен нулю. Определение минимальной (или нулевой) эквипотенциали позволит найти границу раздела между классами.

Заключение

В данной работе была рассмотрена задача распознавания образов и ее частный случай – задача распознавания пешеходов. Были приведены методы решения задачи распознавания. Разработано программное обеспечение на языке C#, реализующее работу трёх наиболее эффективных алгоритмов распознавания изображений: метода опорных векторов, многомерных нейронных сетей и метода потенциальной функции. Решена практическая задача распознавания изображений, а также выполнен анализ полученных результатов.

Список использованных источников

1. Воронцов К. В. Лекции по алгоритмам кластеризации и многомерного шкалирования. М.: МГУ, 2007. 18 с.
2. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. Новосибирск: ИМ СО РАН, 1999. 270 с.
3. Лепский А.Е., Броневиц А.Г. Математические методы распознавания образов. Таганрог:ТТИ ЮФУ, 2009. 155с.
4. Местецкий Л.М. Математические методы распознавания образов. М.:МГУ, 2004. 85с.