Как обучаются машины?

научно-популярная лекция

Н. Ю. Золотых

20 сентября 2008

Аннотация

Рассмотрены реальные задачи из разных областей науки и техники. Приведены их формулировки в виде задач машинного обучения. Кратко описаны некоторые алгоритмы.

Содержание

1. Что такое машинное обучение?

Прогресс в области информационных технологий за последние 20 лет громаден. Особое впечатление производят вещи, которые раньше считались прерогативой исключительно человека: компьютеры научились распознавать рукописный текст и речь, они видят дорогу и управляют автомобилем, они играют в шахматы

на уровне чемпионов мира, компьютеры ставят диагноз больным, они умеют определять в тексте ключевые слова и по ним классифицировать его... Все это было бы, по-видимому, не возможным, если бы человек не научил компьютер *обучаться*. Очень сложно (или даже невозможно) запрограммировать явный алгоритм, по которому компьютер сможет, например, распознавать рукописный текст или речь, но можно запрограммировать алгоритм, по которому он будет обучаться этому.

По всей видимости, термин «машинное обучение» (machine learning) впервые ввел в употребление А. Самуэль в 1959 г. В своей работе «Исследование в области машинного обучения на примере игры в шашки»[1] он неформально определил понятие *машинного обучения* как процесса, в результате которого машина (компьютер) способна показывать поведение, которое в нее не было *явно* заложено (запрограммировано). Самуэль описывает компьютерную программу, умеющую играть в шашки и умеющую обучаться на основе опыта. Его исследования показали следующее:

> Компьютер можно запрограммировать так, что он сможет играть лучше ... человека, написавшего программу. Более того, он может научиться этому за достаточно короткий промежуток времени [игры с самим собой] ... если задать ему только правила игры, понимание цели и некоторый излишний и неполный список параметров ... знаки которых и относительные веса не определены.

Намного позже Т. Митчелл[2] даст более строгое определение термину «машинное обучение»:

> Говорят, что компьютерная программа *обучается* на основе опыта E по отношению к некоторому классу задач T и меры качества P, если качество решения задач из T, измеренное на основе P, улучшается с приобретением опыта E.

Как же научить машины обучаться? Существуют разные подходы. В 1958 г. Ф. Розенблатт[3] построил устройство, названное им *персептрон*, которое технически реализовало простую модель мозга, предложенную ранее физиологами (модель МакКаллоха–Питса). Эта модель состоит из большого числа простых агентов (нейронов), взаимодействующих между собой. Персептрону показывали написанные рукой цифры, а также вводили в него информацию о том, какая именно цифра изображена. После такого обучения персептрон почти без ошибок мог сам классифицировать предлагаемые образы.

Как отмечают многие, успех персептрона рассматривался не только как успех в решении одной конкретной задачи (распознавание графических образов), но и как успех идеи использовать для решения интеллектуальных задач просто организованные системы с большим числом агентов. Некоторым исследователям персептрон представлялся универсальным устройством для решения любых интеллек-

[1] *A.L. Samuel* Some Studies in Machine Learning Using the Game of Checkers // IBM Journal. July 1959. P. 210–229.

[2] *T.M. Mitchell* Machine Learning. McGraw-Hill, 1997.

[3] *F. Rosenblatt* The Perceptron. A Probabilistic Model for Information Storage and Organization in the Brain // Phys. Rev, V. 6, № 65, November, 1958.

туальных задач — нужно только увеличить число нейронов и подсмотреть получше, как устроен человеческий мозг. Возможности современной техники позволяют значительно увеличить количество нейронов, но не в количестве дело. Вапник отмечает[4]:

> Конечно, очень интересно знать, как человек учится. Однако совсем не обязательно, что это лучший путь для построения искусственных самообучающихся машин. Замечено, что исследование полета птиц никак не пригодилось при конструировании самолета.

Мы не можем сказать, что на пути прямого моделирования мозга достигнуты весьма значительные результаты в области машинного обучения, несмотря на реинкарнацию персептрона в середине 80-х годов в виде нейронных сетей. Кроме того, человечеству, по-видимому, еще далеко до конструирования *универсального* решателя интеллектуальных задач, но *конкретные* сложные задачи мы можем научить решать компьютер уже сейчас. При этом, разумеется, находят свое применение классические методы (регрессия и метод наименьших квадратов, статистическая оценка параметров функции распределения, дискриминантный анализ и др.), но, конечно, появляются и бурно развиваются новые подходы (деревья решений, машина опорных векторов, бустинг и др.)

По аналогии с обучением людей мы можем классифицировать типы обучения машин. Выделяют следующие типы обучения:

- *Дедуктивное*, или *аналитическое*, обучение. Имеются знания, сформулированные экспертом и как-то формализованные. Программа должна выводить из этих правил конкретные факты и новые правила.
- *Индуктивное* обучение. На основе эмпирических данных программа строит общее правило. Эмпирические данные могут быть получены самой программой в предыдущие сеансы ее работы или просто предъявлены ей.
- *Комбинированное* обучение, содержащее элементы как дедуктивного, так и аналитического обучения.

Дедуктивное обучение относят к области экспертных систем. Здесь мы будем рассматривать только индуктивное обучение.

В типичном сценарии индуктивного обучения мы имеем *объекты*, каждый из которых характеризуется некоторым набором *признаков*, или *свойств*. Кроме того, каждому объекту приписана отдельная величина, называемая *выходом*. Имеется *обучающая выборка* — конечное множество наблюдаемых объектов, у каждого из которых мы знаем значения всех его признаков и значение выхода. Используя эту выборку, мы должны построить *решающее правило*, которое бы для каждого нового объекта по его признакам предсказывала бы выход. Признаки объекта называются также *входами*. Таким образом, по входам требуется научиться предсказывать выход.

[4] *V. Vapnik* The Nature of Statistical Learning Theory. 2nd ed. Springer, 2000.

Признаки и выходы бывают *количественные*, как, например, цена, температура, и *качественные*, или *номинальные*, как, например, пол, название заболевания, отсутствие/присутствие конкретного симптома и т. п. Признак, принимающий только два значения, называется *бинарным*. Если выход количественный, то восстанавливаемое решающее правило называется *регрессией*, а сама задача — *задачей восстановления регрессии*. Если выход *качественный*, то решающее правило называется *классификатором*, а задача — *задачей классификации*, или *задачей распознавания образов*.

Восстановление регрессии и классификация — это примеры задач *обучения с учителем*, так как для каждого объекта из обучающей выборки известен выход, и можно считать, что его указывает некий учитель. В рамках индуктивного обучения рассматривают также *обучение без учителя*, в котором для объектов обучающей выборки выходы не известны. В этом случае необходимо определить, как объекты связаны друг с другом, например, выделить группы (*кластеры*) близких по своим свойствам объектов.

Далее мы подробно остановимся на каждом их упомянутых типов задач, рассмотрим примеры из практики и познакомимся с некоторыми методами решения задач.

2. Регрессия

2.1. «Регрессия к середине» Ф. Гальтона

В 1885 г. в статье «Регрессия к середине в наследовании роста»[5] Фрэнсис Гальтон (1822–1911) приводит данные о росте группы родителей и их взрослых детей. Анализируя эти данные, Гальтон приходит к выводу, что дети не проявляют тенденции к сохранению роста их родителей. Наоборот, их рост в среднем оказывался ближе к середине, чем у родителей. К аналогичным выводам он приходит, анализируя данные о размере семян: семена были, как правило, меньше, если родители были велики, и больше, если родители были малы. Эту сыновнюю тенденцию к среднему значению Гальтон назвал *регрессией к середине*. После работ Гальтона термин «регрессия» (буквально: возвращение, движение назад) стал использоваться разными биологами, но начиная с трудов К. Пирсона, он повсеместно употребляется в математике для обозначения зависимостей с количественным выходом.

Данные Гальтона включают 928 пар

$$(x_1,\, y_1),\ (x_2,\, y_2),\ \ldots\ ,\ (x_N,\, y_N), \tag{1}$$

где x_i — средний рост двух родителей, а y_i — рост их взрослого ребенка[6], $N = 928$. *Диаграмма рассеяния* для данных Гальтона представлена[7] на рис. 1.

[5] *F. Galton* Regression towards Mediocrity in Hereditary Stature // Journal of the Antropological Institute. 1885, 15. P. 246–263.

[6] Для учета того, что мужчины в среднем выше женщин, использовались поправочные коэффициенты.

[7] У Гальтона данные приведены с 1 знаком после запятой. В этом случае многие точки совпали бы друг на другом. Чтобы они выглядели как «облако», мы добавили к координатам каждой точки малое случайное возмущение.

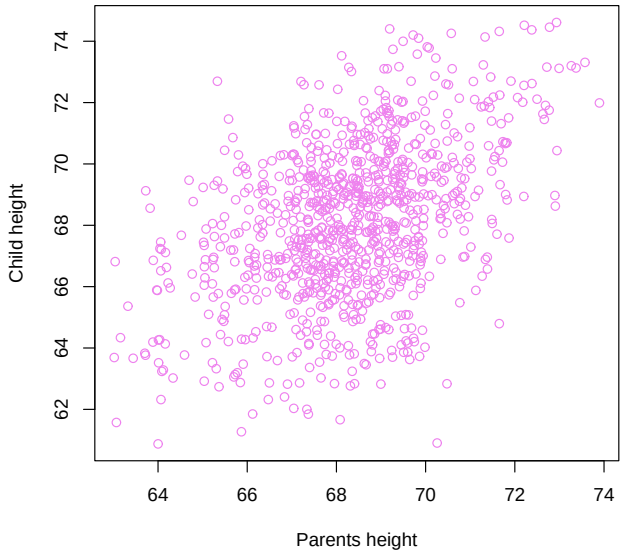

Рис. 1. Диаграмма рассеяния для данных из исследования Ф. Гальтона. По горизонтальной оси откладывается рост родителей x. По вертикальной — рост детей y.

Попробуем сами проанализировать эти данные и получить аналогичный результат.

Из рис.1 видно, что точки располагаются вдоль некоторой прямой

$$y = \beta_0 + \beta_1 x, \tag{2}$$

где β_0, β_1 — пока не известные коэффициенты. Таким образом,

$$y_i \approx \beta_0 + \beta_1 x_i \qquad (i = 1, 2, \ldots, N).$$

Если бы β_0, β_1 были известны, то по формуле (2) мы могли бы предсказывать значение y по заданному x. Как правило, мы будем совершать некоторую ошибку, но тенденция будет угадана правильно. Рассмотрим *остаточную сумму квадратов*, равную

$$R(\beta_0,\, \beta_1) = \sum_{i=1}^{N} (y_i - \beta_0 - \beta_1 x_i)^2.$$

Здесь $y_i - \beta_0 - \beta_1 x_i$ равно отклонению предсказанного значения $\beta_0 - \beta_1 x_i$ от истинного y_i и, таким образом, $R(\beta_0,\, \beta_1)$ характеризует общую ошибку на имеющихся данных.

В качестве β_0, β_1 выберем значения, на которых достигается минимум функции $R(\beta_0,\, \beta_1)$. Находим частные производные:

$$\frac{\partial R}{\partial \beta_0} = -\sum_{i=1}^{N} 2(y_i - \beta_0 - \beta_1 x_i), \qquad \frac{\partial R}{\partial \beta_1} = -\sum_{i=1}^{N} 2x_i(y_i - \beta_0 - \beta_1 x_i).$$

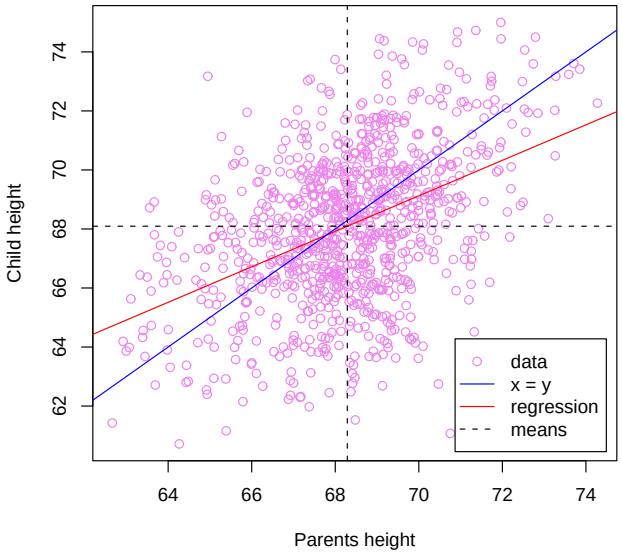

Рис. 2. Зависимость роста взрослого ребенка от роста родителей в исследовании Ф. Гальтона. По горизонтальной оси откладывается рост родителей x. По вертикальной — рост детей (в дюймах) y. Синяя прямая имеет уравнение $y = x$. Красная прямая построена по методу наименьших квадратов по эмпирическим данным. Средние значения обозначены пунктирными линиями. Взаимное расположение красной и синей прямой показывает, что рост детей имеет тенденцию (регрессию) к середине.

Приравнивая их к нулю и делая очевидные преобразования, получаем *систему нормальных уравнений*

$$\begin{cases} \beta_0 + \beta_1 \sum\limits_{i=1}^{N} x_i = \sum\limits_{i=1}^{N} y_i, \\ \beta_0 \sum\limits_{i=1}^{N} x_i + \beta_1 \sum\limits_{i=1}^{N} x_i^2 = \sum\limits_{i=1}^{N} x_i y_i, \end{cases} \tag{3}$$

относительно неизвестных β_0, β_1. Система имеет решение $\beta_0 = 24$, $\beta_1 = 0.65$. Таким образом, восстановленная зависимость имеет вид

$$y = 24 + 0.65x. \tag{4}$$

График восстановленной зависимости приведен на рис. 2.

Метод, с помощью которого мы нашли β_0, β_1, называется *методом наименьших квадратов*. Он был разработан К. Гауссом, Р. Эдрейном и А. Лежандром на рубеже XVIII–XIX вв.

Зависимость (4) можно записать следующим образом (проверьте!):

$$y \approx 68.2 + 0.65 \cdot (x - 68.2) \tag{5}$$

Здесь 68.2 приближенно равно среднему значению для x_i и для y_i:

$$\frac{1}{N}\sum_{i=1}^{N} x_i \approx \frac{1}{N}\sum_{i=1}^{N} y_i \approx 68.2.$$

Проанализировав формулу (5) и график восстановленной зависимости на рис. 2, вслед за Гальтоном мы можем сказать, что «сыновняя регрессия к середине оказалась прямо пропорциональной отклонению родителей от нее».

В терминологии машинного обучения последовательность (1) — это обучающая выборка, x_i — объекты, а y_i — выходы. Каждый объект характеризуется только одним признаком (средний рост родителей). И вход, и выход в этой задаче были количественными. Функция (4) называется регрессией, а коэффициенты β_0, β_1 — *регрессионными коэффициентами*. Таким образом, мы решили *задачу восстановления регрессии*.

Как уже отмечалось, термин «регрессия» после работ Гальтона стал употребляться для обозначения методов изучения зависимостей в совершенно других ситуациях и со временем произошло некоторое переосмысление этого термина. Понятие «регрессия» (движение назад) стало употребляться в значении, близком к термину «индукция» (переход от частного к общему). В исследовании Гальтона по «сырым» данным определяется (восстанавливается $\equiv$ движение назад) общее правило. Такой метод в корне отличается от традиционного для того времени подхода, когда данные только подтверждали закон.

2.2. Как определить цену дома? Множественная регрессия

Предположим, вы хотите продать дом или квартиру и желаете определить, какую назначить цену. Пусть вам доступна база данных, содержащая информацию о жилье, включая цену, состояние, жилую площадь, количество этажей, количество комнат, время постройки, удаленность до основных магистралей, наличие инфраструктуры, экологическую обстановку в районе и т. п. Как, пользуясь всей этой информацией, определить цену вашего жилья?

Эта задача есть задача восстановления регрессии. Объектами являются дома. Признаками — их характеристики. Выход — цена дома.

Для примера рассмотрим Boston Housing Data[8]. База содержит информацию о загородных домах близ Бостона. Данные были собраны в 1970-х годах. Информация агрегирована: территория поделена на участки и дома, стоящие на одном участке, собраны в группы. Таким образом, объектами являются сами эти группы. Их общее количество — 506. В качестве признаков рассматриваются:

1. `CRIM` — уровень преступности на душу населения,
2. `ZN` — процент земли, застроенной жилыми домами (только для участков площадью свыше 25000 кв. футов),
3. `INDUS` — процент деловой застройки,

[8] См. UCI Repository of Machine Learning Databases `http://www.ics.uci.edu/~mlearn/MLRepository.html`

4. CHAS — 1, если участок граничит с рекой; 0 в противном случае (бинарный признак),
5. NOX — концентрация оксида азота, деленная на 10^7,
6. RM — среднее число комнат (по всем домам рассматриваемого участка),
7. AGE — процент домов, построенных до 1940 г. и занимаемых владельцами,
8. DIS — взвешенное расстояние до 5 деловых центров Бостона,
9. RAD — индекс удаленности до радиальных магистралей,
10. TAX — величина налога \$10000,
11. PTRATIO — количество учащихся, приходящихся на одного учителя (по городу),
12. B = $1000(\mathtt{AA} - 0.63)^2$, где AA — доля афро-американцев,
13. LSTAT — процент жителей с низким социальным статусом.

Признак CHAS — бинарный, остальные — количественные. Выходом является переменная MEDV, равная медианному значению цены строения (по всем домам участка) в \$1000.

Судить о корреляции переменных можно по диаграммам рассеяния. На рис. 3 представлены диаграммы рассеяния для каждой пары переменных MEDV, INDUS, NOX, RM, AGE, PTRATIO, B. Изображены только по 100 точек, случайно выбранных из данных.

Попытаемся найти зависимость между входом и выходом в виде функции (регрессии)

$$\mathtt{MEDV} = \beta_0 + \beta_1 \cdot \mathtt{CRIM} + \beta_2 \cdot \mathtt{ZN} + \ldots + \beta_{13} \cdot \mathtt{LSTAT}.$$

или

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p \cdot x_p, \tag{6}$$

где $p = 13$, а переменные MEDV, CRIM, ZN, ..., LSTAT обозначены соответственно через y, x_1, $x_2, \ldots, x_p$. Выход зависит не от одной, а от 13 входных переменных, поэтому регрессия называется *множественной*. Теперь нужно определить 14 неизвестных параметров, но подход, рассмотренный ранее, будет работать и в новом случае. Остаточная сумма квадратов теперь составляет

$$R(\beta_0, \beta_1, \ldots, \beta_p) = \sum_{i=1}^{N} (y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \ldots - \beta_p x_{ip})^2,$$

где $y_i, x_{i1}, x_{i2}, \ldots, x_{ip}$ — значения переменных y, x_1, $x_2, \ldots, x_p$ соответственно для i-го объекта из обучающей выборки (i-го участка в базе). Взяв частные производные по параметрам $\beta_0, \beta_1, \ldots, \beta_p$ и приравняв их к нулю, приходим к системе уравнений, аналогичной (3), из которой определяем значения параметров.

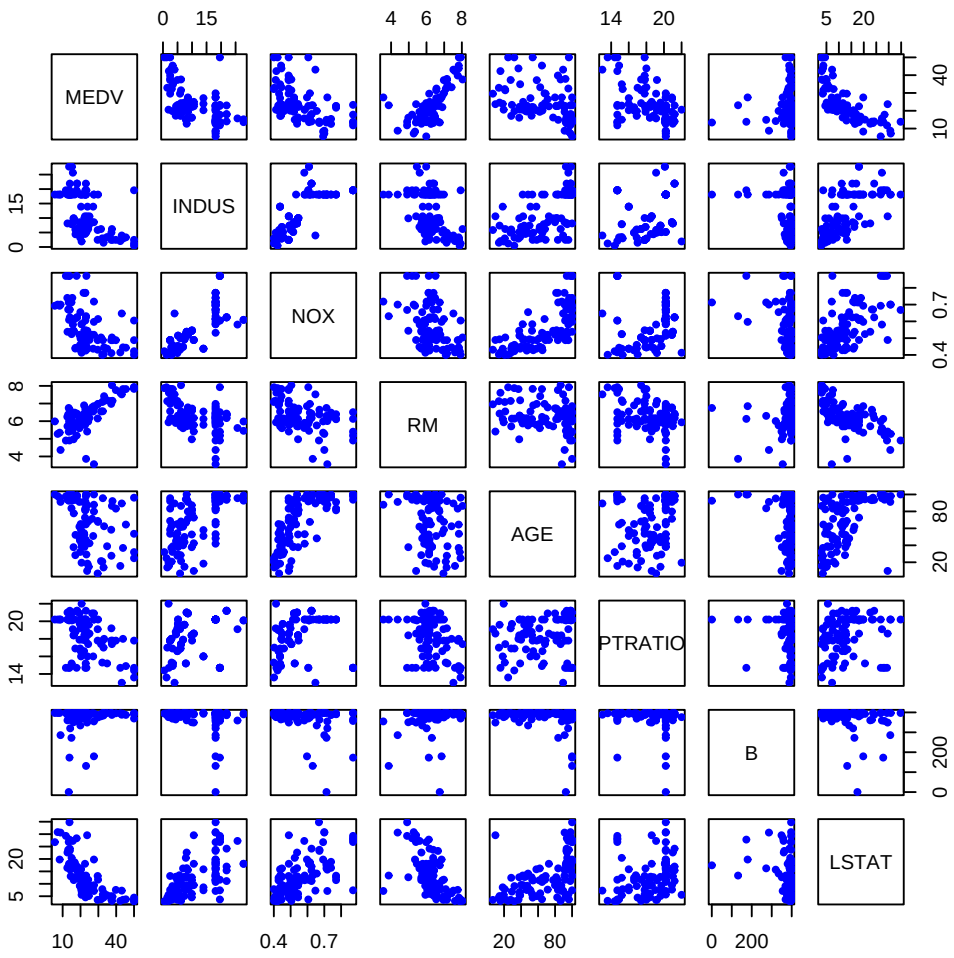

Рис. 3. Диаграммы рассеяния для каждой пары переменных `MEDV`, `INDUS`, `NOX`, `RM`, `AGE`, `PTRATIO`, `B` в задаче о предсказании стоимости дома. Значение переменной `MEDV` нужно научиться предсказывать по значениям остальных переменных.

3. Классификация

3.1. Как предсказать уровень гормона?

На рис. 4 представлены данные о 114 лицах с заболеванием щитовидной железы[9]. На момент проведения обследования у 61 из них был повышенный уровень свободного гормона T4 (гиперфункция щитовидной железы), у 53 пациентов уровень этого гормона был в норме. Для каждого пациента известны следующие показатели: `ЧСС` — частота сердечных сокращений (пульс), и `SDNN` — стандартное отклонение длительности интервалов между синусовыми сокращениями RR. Из-

[9] Данные представлены врачом больницы 13 г. Нижнего Новгорода М.Н. Будкиной.

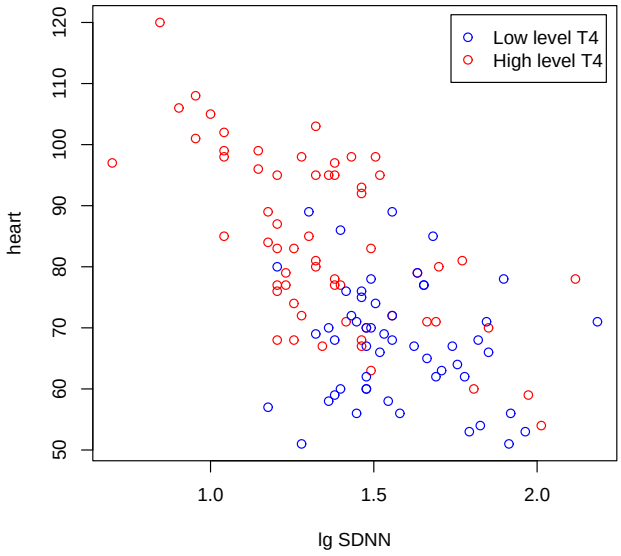

Рис. 4. Диаграмма, представляющая данные о группе лиц с заболеванием щитовидной железы. По горизонтальной оси отложен десятичный логарифм от SDNN. По вертикальной — ЧСС. Синие маркеры соответствуют пациентам с нормальным уровнем, а красные — пациентам с повышенным уровнем T4.

мерение ЧСС и SDNN проще и дешевле установления уровня T4, поэтому возникает вопрос, можно ли научиться предсказывать (допуская небольшие ошибки) уровень свободного T4 по ЧСС и SDNN.

На рис. 5 изображены те же точки и, кроме того, прямая[10] с уравнением

$$16 \cdot \lg \mathtt{SDNN} - \mathtt{ЧСС} + 50 = 0, \tag{7}$$

разбивающая плоскость на две области. Мы видим, что она достаточно удачно разбивает множество точек, так, что по одну ее сторону находятся преимущественно синие точки, а по другую — преимущественно красные. Таким образом, если к нам поступит новый пациент, для которого известны SDNN и ЧСС, мы должны определить, по какую сторону от прямой находится соответсвующая его показателям точка и на основании этого предсказать, какой у него уровень T4. На языке алгебры это означает следующее. Для заданных SDNN и ЧСС необходимо вычислить функцию

$$f(\mathtt{SDNN}, \mathtt{ЧСС}) = 16 \lg \mathtt{SDNN} - \mathtt{ЧСС} + 50.$$

Решающее правило заключается в следующем. Если $f(\mathtt{SDNN}, \mathtt{ЧСС}) > 0$, то у пациента нормальный уровень T4, а если $f(\mathtt{SDNN}, \mathtt{ЧСС}) < 0$, то уровень T4 повышенный. Данное решающее правило совершает ошибку в 23 % случаев.

[10] Линия с уравнением (7) будет прямой в плоскости $\log \mathtt{SDNN}$, ЧСС. Если перейти от логарифма к исходной величине SDNN, то эта линия уж не будет прямой.

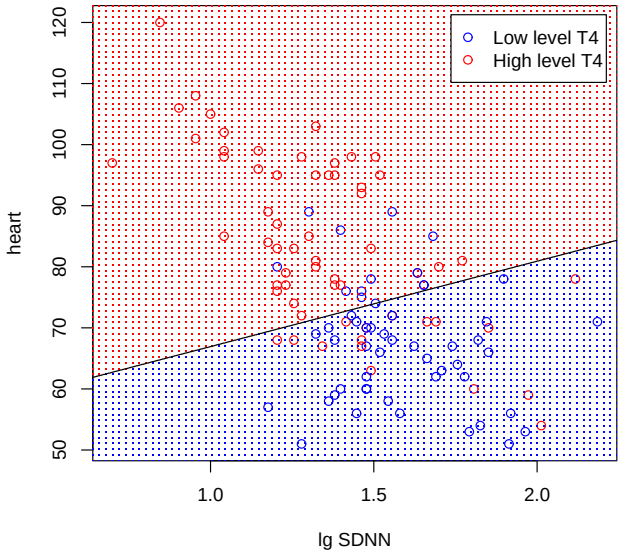

Рис. 5. Прямая разделяет плоскость на две области и тем самым определяет решающее правило.

Рассмотренная задача — пример задачи *классификации*. По обучающей выборке

$$(x_1, y_1),\ (x_2, y_2),\ \ldots, (x_N, y_N),$$

где x_i — пациент из базы данных, а y_i — класс, к которому он принадлежит (например, $y_i = 0$ означает, что уровень T4 у него в норме, $y_i = 1$ — повышенный уровень), мы научиться предсказывать, к какому классу будут принадлежать новые пациенты, т. е. построить решающее правило. Таким образом, задача классификации похожа на задачу восстановления регрессии и отличается от нее тем, что выход может принимать лишь конечное число значений (в данном случае два). В задаче классификации решающее правило называется *классификатором*.

В рассмотренной задаче каждый пациент характеризовался двумя признаками: `SDNN` и `ЧСС`, т. е. пространство признаков было двумерным. Нам удалось найти прямую, которая задает достаточно хорошее решающее правило. В трехмерном пространстве аналогом прямой является плоскость, а в пространствах бóльшей размерности — гиперплоскость. Гиперплоскость задается уравнением вида

$$\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p = 0$$

и разбивает пространство на две части. Для одной из них $\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p < 0$, а для другой $\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p > 0$. Достаточно часто хорошее решающее правило удается задать с помощью разделяющей гиперплоскости. Иногда для этого от исходных входных переменных нужно перейти к некоторым функциям от них. Именно так мы поступили в предыдущем примере: вместо `SDNN` рассматривали $\lg$ `SDNN`.

3.2. Обобщающая способность решающего правила

Построенное нами в предыдущем разделе решающее правило на обучающей выборке давало ошибку в 23 %. Понятно, что эту ошибку можно было бы сделать меньше, если провести не прямую, а более замысловатую разделяющую линию, или, тем более, задать классы с помощью несвязных областей. Однако это вовсе не означало бы, что таким образом мы станем лучше классифицировать *новые* объекты.

Чтобы прочувствовать это рассмотрим следующий экстремальный пример. В некоторой задаче классификации с объектами из двух классов определим следующее решающее правило. Объекты из обучающей выборки будем классифицировать правильно, а все новые объекты будем случайно равновероятно относить к произвольному классу. Очевидно, что ошибка на тестовой выборке равна 0 %. При этом на новых объектах в среднем она будет составлять 50 % (если объекты из каждого класса появляются равновероятно).

Итак, *малая ошибка на данных, по которым построено решающее правило, не гарантирует, что ошибка на новых объектах также будет малой*. *Обобщающая способность* (*качество*) *решающего правила* — это способность решающего правила правильно предсказывать выход для новых объектов, не вошедших в обучающую выборку.

Как же оценить, насколько хорошо решающее правило будет классифицировать новые объекты, т. е. как оценить качество решающего правила? Обычно поступают следующим образом. Имеющиеся данные случайно разбивают на два подмножества. Одно используют как *обучающую выборку* и строят по нему решающее правило. Другое называют *тестовой* или *контрольной выборкой* и используют для оценки уровня ошибки на новых данных. Этот подход мы разберем далее на примере.

3.3. Машина опорных векторов

Познакомимся с основами знаменитого метода построения решающего правила, носящего название «*машина опорных векторов*». Этот метод был предложен В.Н. Вапником и А.Я. Червоненкисом в 1974 г.[11] Познакомимся с основами этого метода.

На рис. 6 представлены два класса точек и разделяющая прямая. Классы точно разделяются этой прямой, так, что по одну сторону от нее располагаются точки из первого класса, а по другую — из второго. Понятно, что таких разделяющих прямых бесконечно много. Мысленно сместим параллельно прямую вначале в одну, а затем в другую сторону, пока прямая не коснется какой-либо точки из обучающей выборки. Область, которую при этом заметет прямая, окрашена на рис. 6 в желтый цвет. Назовем эту область *нейтральной полосой*.

Среди бесконечного множества возможных разделяющих прямых выберем ту, для которой ширина нейтральной полосы максимальна и при этом прямая лежит в точности посередине этой полосы; см. рис. 7. Назовем эту прямую *оптимальной*.

[11] *В.Н. Вапник, А.Я. Червоненкис* Теория распознавания образов. М.: Наука, 1974. Название «машина опорных векторов» метод приобрел позднее.

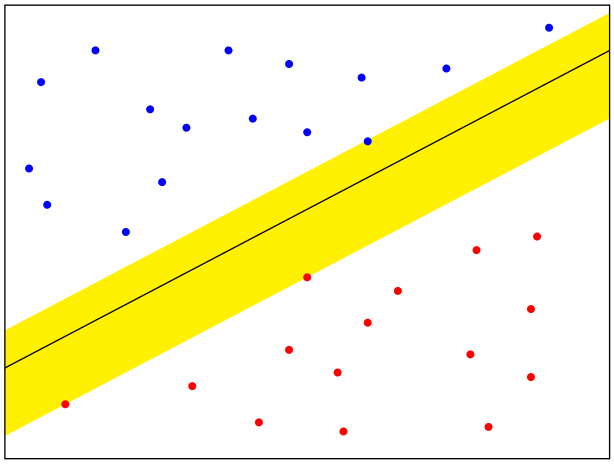

Рис. 6. Прямая точно разделяет множество на два заданных класса. Нейтральная полоса закрашена желтым цветом.

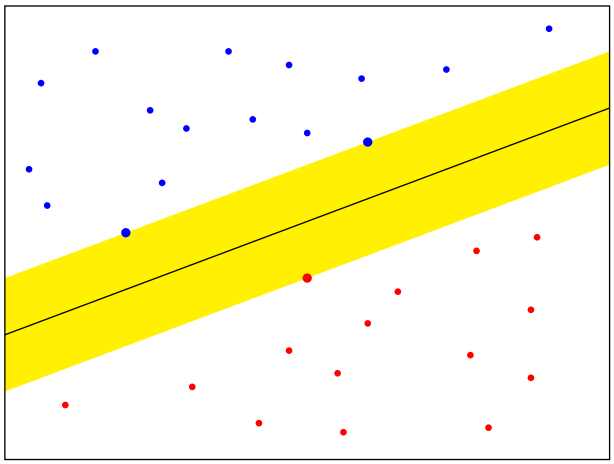

Рис. 7. Оптимальная разделяющая прямая. Нейтральная полоса имеет максимальную ширину и разделяющая прямая лежит в точности посередине ее.

Пусть решающее правило определяется оптимальной разделяющей прямой. Интуитивно ясно, что лишь небольшое число точек, соответсвующих новым объектам, будет попадать в нейтральную полосу и совсем малое их число перескочет на «чужую территорию».

Аналогичным образом вводится понятие оптимальной гиперплоскости в p-мерном пространстве признаков. Задача поиска оптимальной гиперплоскости сводится к задаче квадратичного программирования (минимизации квадратичной функции при линейных ограничениях), для решения которой разработаны хорошие чис-

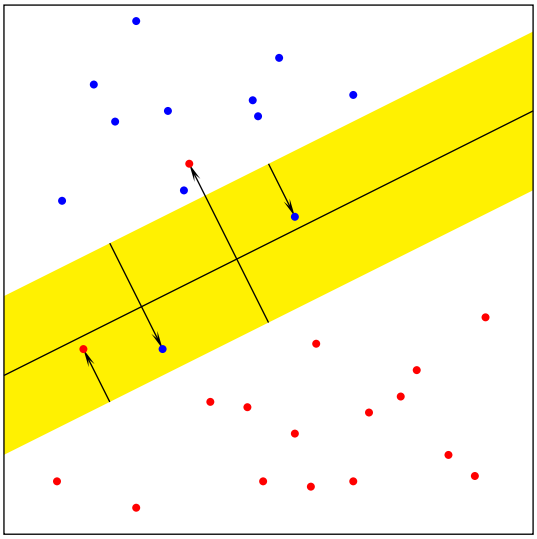

Рис. 8. Точки нелзя разделить прямой на два заданных класса. Точки, которые попали на нейтральную полосу, или зашли на «чужую территорию» отмечены стрелками.

ленные методы.

Что делать, если точки нельзя разделить гиперплоскостью на два заданных класса? В этом случае решают несколько измененную задачу. В ее формулировке фигурирует некоторый настроечный параметр, с помощью которого можно регулировать количество точек из обучающей выборки, которые попадут в нейтральную полосу или окажутся на «чужой территории»; см. рис. 8.

3.4. Распознавание рукописных символов

Рассмотрим задачу распознавания символов. Выборка `optdigit`[12] содержит информацию о 1934 изображениях рукописных цифр, некоторые из них представлены на рис. 9. Каждое изображение закодировано вектором длины 1024, составленном из нулей и единиц. Каждая компонента вектора соответствует своему пикселу на изображении размера 32×32 и равна 1, если пиксел подсвечен и 0 в противном случае. Все изображения перед кодированием были отмасштабированы так, чтобы они имели примерно одинаковые размеры. Для каждого изображения известно, какое число на нем представлено (т. е. известен класс). Получили задачу классификации, при этом объектами являются изображения, входами — бинарные признаки $x_1, x_2, \ldots, x_{1024}$, а выходом — класс.

На этой выборке автор испытал несколько алгоритмов классификации. Предварительно исходная выборка была случайно поделена на две части по 967 объектов в каждой. Первая выборка (обучающая) использовалась для обучения, т. е. настройки параметров классификаторов. На второй выборке (контрольной) путем

[12]UCI Repository of Machine Learning Databases `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

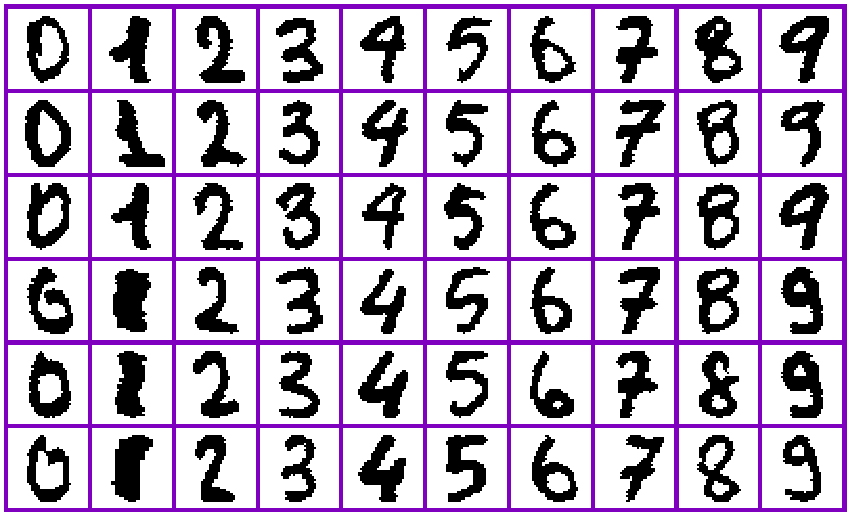

Рис. 9. Некоторые объекты из обучающей выборки в задаче `optdigit` распознавания рукописных цифр.

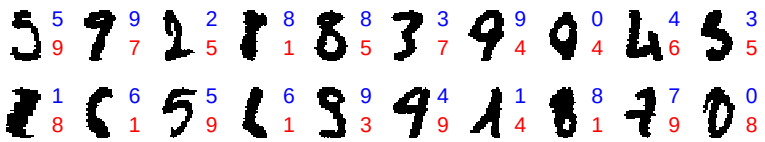

Рис. 10. Изображения, не правильно распознанные с помощью машины опорных векторов. Синяя цифра в правом верхнем углу изображения — правильный ответ. Красная цифра в правом нижнем углу каждого изображения — ответ классификатора.

вычисления ошибки оценивалась его обобщающая способность. Ошибки на обучающей и тестовой выборках приведена в следующей таблице.

Алгоритм	*Ошибка*	
	на обучающей выборке	*на тестовой выборке*
Машина опорных векторов	0 %	2.1 %
Метод ближайшего соседа	0 %	3.1 %
Нейронная сеть	0 %	4.7 %
AdaBoost	0 %	4.8 %

Как видим, лучшие результаты показала машина опорных векторов. Все случаи неправильной классификации цифр из тестовой выборки при использовании машины опорных векторов приведены на рис. 10.

Рис. 11. Диаграмма, представляющая данные о времени извержения и промежутках между извержениями гейзера.

4. Кластеризация

4.1. Извержение гейзера

На рис. 11 представлены данные о времени между извержениями и длительностью извержения Old Faithful geyser in Yellowstone National Park, Wyoming, USA[13] Мы видим, что точки группируются в два кластера. В одном кластере находятся точки, соответствующие извержениям с малой длительностью и малым временем ожидания. В другом — с большой длительностью и большим временем ожидания.

Это задача *кластеризации*. Здесь, как и в задаче классификации и задаче восстановления регрессии, имеются объекты (извержения гейзера), обладающие некоторыми признаками (длительность и промежуток между извержениями), однако не задан выход. Требуется выделить кластеры — группы объектов со схожими (близкими) признаками. Так как выход не известен, то задачу кластеризации относят к классу *задач обучения без учителя*.

На рис. 12 кластеры выделены. Также отмечены центры тяжестей в каждом классе. Для нахождения центра тяжести необходимо найти среднее значение каждого признака.

Рассмотрим *метод центров тяжести* для решения задачи кластеризации. Пусть множество объектов требуется разбить на K кластеров. Вначале произвольным

[13] *A. Azzalini, A.W. Bowman* A look at some data on the Old Faithful geyser // Applied Statistics. 1990, 39. P. 357-–365.

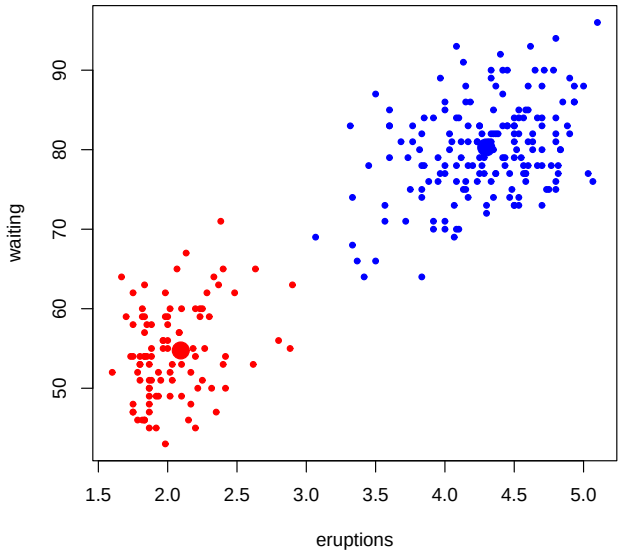

Рис. 12. Данные разбиты на два класса. Жирными точка отмечены центры тяжести кластеров.

образом разобьем объекты на K классов. В каждой группе найдем центр тяжести и проведем перегруппировку. Для этого поместим каждый объект в тот класс, к прежнему центру тяжести которого он ближе всех. После перегруппировки всех объектов снова вычислим центры тяжести и повторим итерацию. Итерации повторяются до тех пор, пока разбиение на классы перестанет изменяться.

К описанному методу близок *метод медиан*. Основное отличие метода в том, что вместо центров тяжести используются медианы. Медианой заданного набора точек называется та из них, которая ближе всех к центру тяжести. Кластеры, представленные на рис. 12 получены методом медиан.

4.2. Далекие миры

Экзопланета — это планета из другой звездной системы. Первая экзопланета была открыта в 1995 г. С тех пор нам стало известно о нескольких сотнях экзопланет. Оказывается, что открытые планеты совсем не похожи на 8 планет Солнечой системы. Рассмотрим данные[14] о 101 планете. Для каждой указаны следующие характеристики (признаки): x_1 — масса (в массах Юпитера), x_2 — период обращения вокруг собственной оси (в земных днях), x_3 — эксцентриситет. Таким образом, каждую планету мы можем представить точкой в 3-мерном пространстве признаков; см. рис. 13.

Попробуем проанализировать эти данные, разбив планеты на группы. Визуально группу точек в трехмерном пространстве уже не столь просто разбить, поэтому

[14] *M. Mayor, P. Frei* New Worlds in the Cosmos: The Discovery of Exoplanets. Cambridge University Press, 2003.

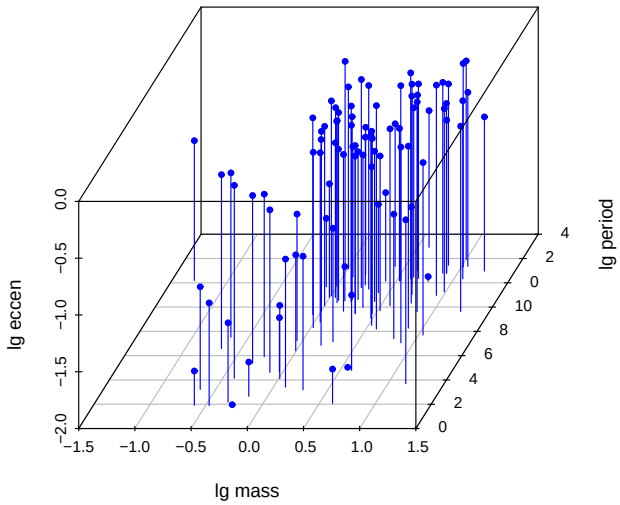

Рис. 13. Диаграмма рзброса векторов признаков для 101 экзопланеты.

воспользуемся каким-либо алгоритмом кластеризации. На рис. 14, 15 представлены результаты работы метода медиан при разбиении на 3 кластера. Выделенные кластеры качественно можно описать следующим образом:

- группа планет с малой массой, малым периодом и разным экцентриситетом;
- группа планет со средней массой и большими периодом и эксцентриситетом;
- группа планет с большими массой, периодом и эксцентриситетом.

Некоторые планеты не совсем соответствуют этим признакам, но их немного.

4.3. Экспрессия генов

Экспрессия — это процесс перезаписи информации с гена на РНК, а затем на белок. Количество и даже свойства получаемого белка зависят не только от гена, но также и от различных внешних факторов (например, от введенного лекарства). Таким образом, уровень экспрессии — это мера количества генерируемого белка (и скорости его генерирования). Для измерения уровня экспрессии генов в имеющемся материале в настоящее время используют миниатюрные приборы, называемые *биочипами* (biochip, microarray).

Каждый биочип предназначен для анализа только вполне определенных генов. В современных устройствах анализируются несколько десятков тысяч разных генов и, например, для анализа всего генетического материала человека достаточно только 3 таких биочипов. Биочип, на который помещается исследуемый генетический материал выглядит как матрица, составленная из микроскопических (диаметра порядка 0.2 мм) точек, светящихся с разной интенсивностью или окрашенных в разные оттенки одного цвета. Чтобы получить свечение, перед помещением на

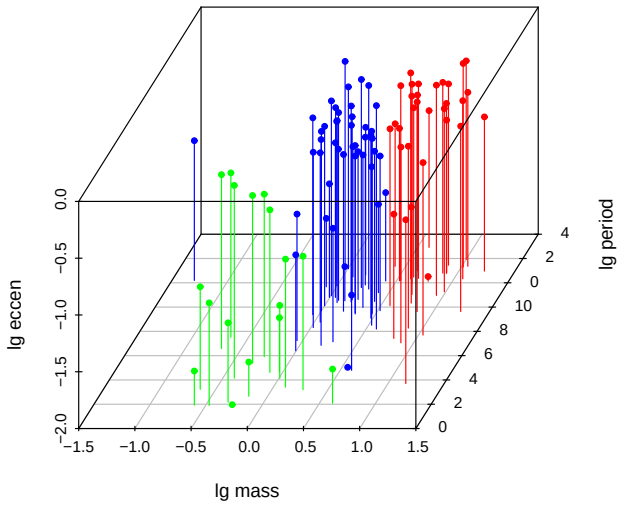

Рис. 14. Результат работы алгоритма медиан при разбиении множества на 3 кластера.

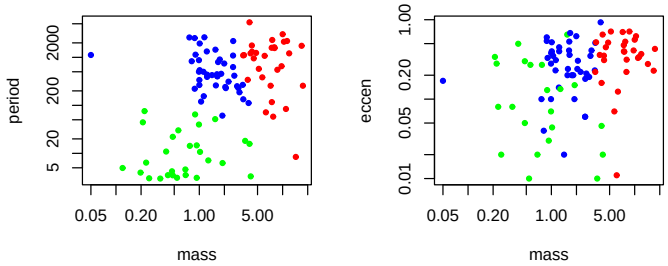

Рис. 15. Результат работы алгоритма медиан при разбиении множества на 3 кластера. Проекции на две координатные плоскости.

биочип к генам прикрепляются молекулы флюоресцентного вещества. Светимость точки пропорциональна уровню экспрессии соответсвующего гена. Если точка не светится, то заданного гена в исследуемом материале нет. Информацию можно представить в виде длинного числового вектора. Каждая его компонента соответствует определенному гену. Значение, хранящееся в этой компоненте, (т. е. яркость точки) указывает на уровень экспрессии гена.

В другом варианте на биочип кроме исследуемого материала помещается также «контрольный» генетический материал. Компоненты получаемого на выходе вектора указывают, как изменился уровень экспрессии генов по сравнению с уровнем экспрессии контрольного генетического материала. Положительные значения соответсвуют увеличению этого уровня по сравнению с контрольным. Отрицательные значения — уменьшению. Перед помещением исследуемый и контрольный генетический материал необходимым образом обрабатывается. В частности,

к материалам прикрепляются молекулы двух разных флюоресцентных веществ, что приводит к окрашиванию точек матрицы в разные оттенки двух цветов. Например, если используются Cy3 (зеленый) и Cy5 (красный), то получим цвета от ярко-зеленого до ярко-красного. Зеленый цвет означает, что выше уровень экспрессии соответствующего гена контрольного материала, красный — что выше уровень экспрессии у исследуемого материала, серый — уровень экспрессии одинаков. Эту информацию можно закодировать в виде числового вектора. Его положительные, отрицательные и нулевые значения означают, что уровень экспрессии у исследуемого материала соответсвенно выше, ниже контрольного уровня или равен ему. Например, если исследуемый материал получен из исходного введением какого-либо лекарства, то положительные значения соответствуют повышению уровеня экспрессии, отрицательные — уменьшению, нулевые означают, что уровень не изменился.

Условное изображение некоторого биочипа[15] приведено на рис. 16. Анализируется генетический материал из пораженных болезнью Паркинсона клеток мозга мыши. Биочип выглядит как матрица размера 132×72. Каждая точка на рисунке соответсвует определенному гену. Таким образом, всего анализируется $132 \times 72 = 9504$ гена. Красный цвет показывает, что уровень экспрессии гена больной клетки выше нормы, а зеленый — ниже нормы. В частности, максимальный уровень экспресии (по сравнению с контрольным уровнем) показывает ген `Gapd` (ярко-красная точка), а минимальный — ген `AA395996` (ярко-зеленая точка).

Пусть было проведено несколько экспериментов, в которых на биочип вместе с контрольным материалом размещались разные другие генетические материалы, например, полученные после введения разных лекарств. Информацию, полученную в результате проведения такой серии экспериментов можно представить в виде числовой матрицы, в которой строки соответсвуют разным генам, а столбцы — разным экспериментам (разным клеткам). Рассмотрим следующие задачи[16]:

(а) Разбить гены на группы в зависимости от влияния на них экспериментов. Гены, реагирующие «почти одинаковым» образом в «большом» числе эсперементов, должны попасть в одну группу. Гены, реагирующие по-разному, должны находиться в разных группах.

(б) Разбить эксперименты на группы в зависимости от их влияния на гены. Эксперименты, в которых одинаковые гены «в основном» реагировали сходным образом должны оказаться в одной группе. Эксперименты, в которых гены реагировали весьма различно, должны находиться в разных группах.

Это задачи кластерного анализа.

В качестве примера рассмотрим данные, полученные исследовательской группой «Genomics Bioinformatics Group»[17]. Имеется информация о 1375 генах в 60 различных клетках. Графически данные представлены в виде матрицы на рис. 17.

[15] Данные взяты из *V.M. Brown, A.Òssadtchi, A.H. Khan, S. Yee, G. Lacan, W.P. Melega, S.R. Cherry, R.M. Leahy, D.J. Smith* Multiplex three dimensional brain gene expression mapping in a mouse model of Parkinson's disease // Genome Research, 2002, 12 (6). P. 868–884.

[16] *T. Hastie, R. Tibshirani, J. Friedman* The elements of statistical learning. Springer, 2001.

[17] `http://discover.nci.nih.gov/datasetsNature2000.jsp`.

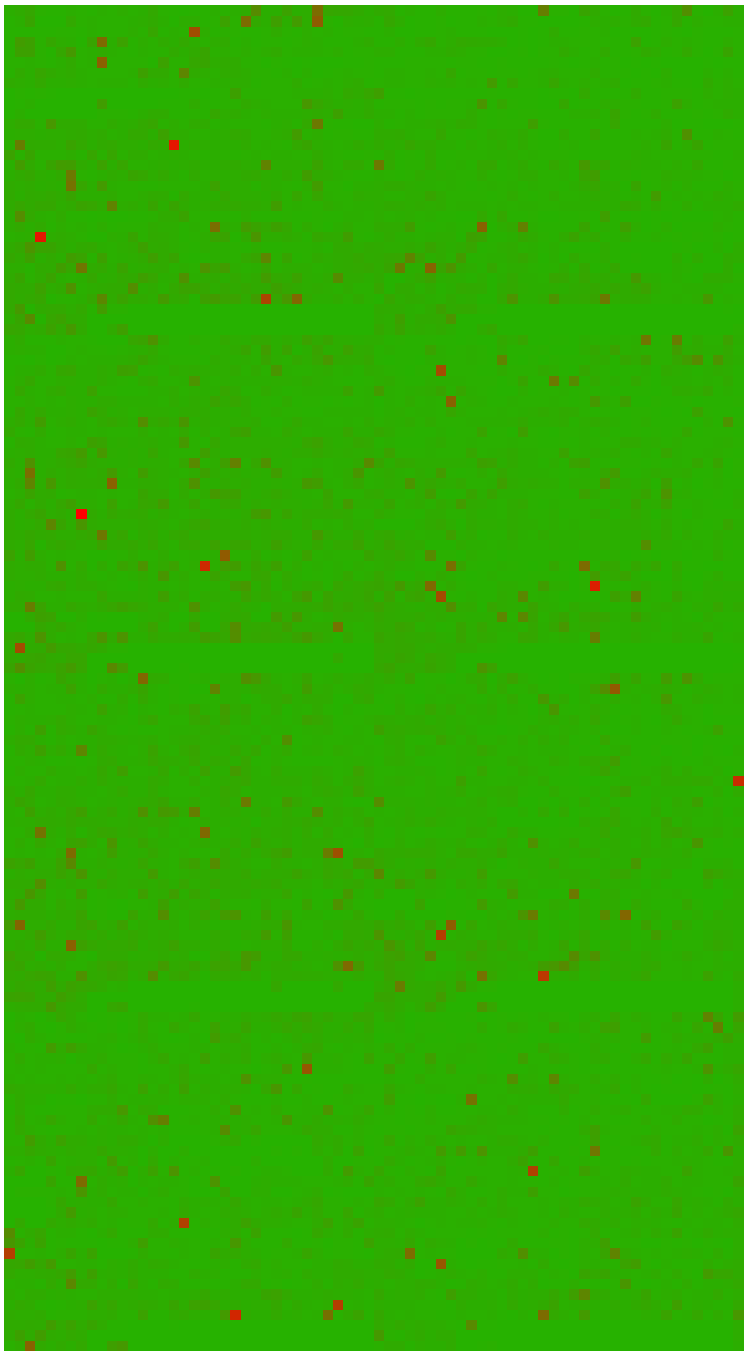

Рис. 16. Условное изображение биочипа. Каждая точка на рисунке соответствует определенному гену. Всего анализируется 9504 гена.

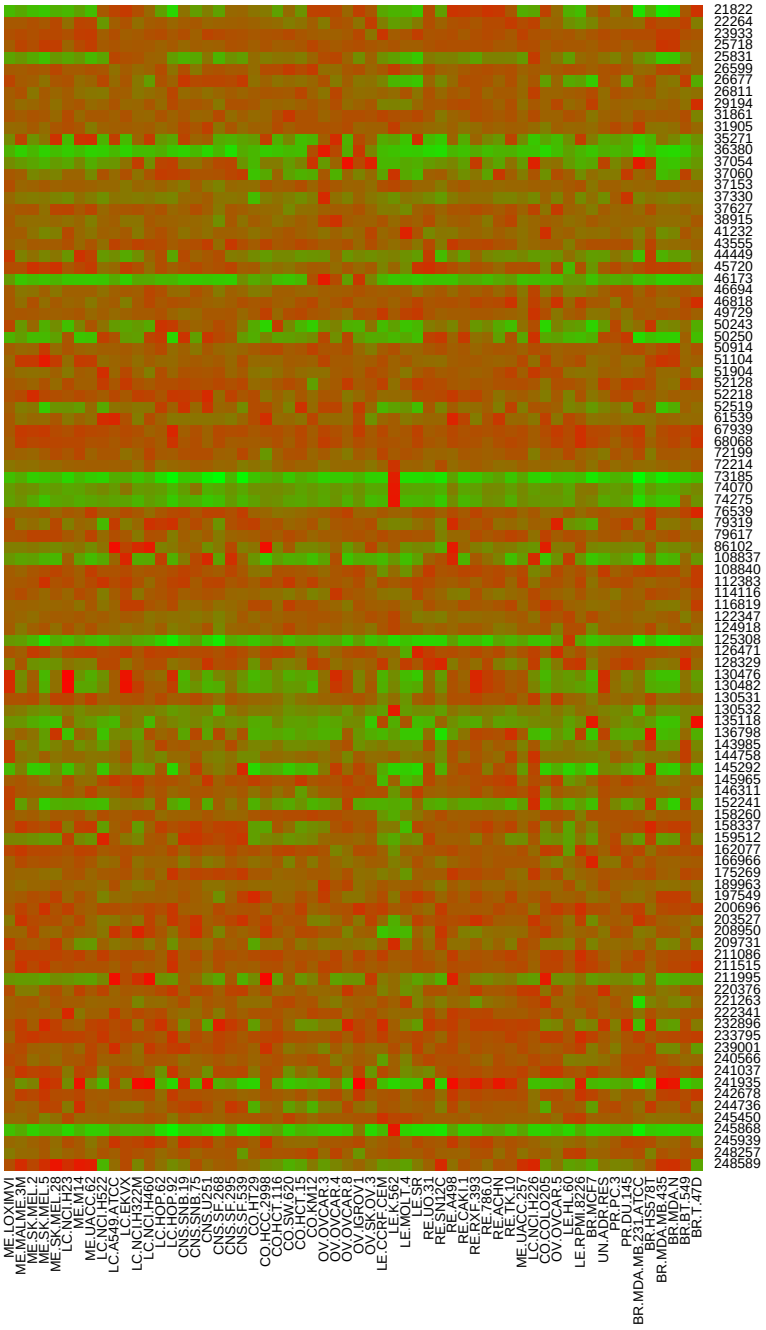

Рис. 17. Данные для 60 экспериментов с биочипом в примере `genome`. Строки соответствуют генам, столбцы — экспериментам. Приведены только первые 100 строк (из общего числа 1375). Строки, содержащие отсутствующие значения, исключены.

Каждая строка соответствует гену, причем изображены только первые 100 строк. Строки, содержащие отсутствующие значения, исключены.

4.4. Иерархическая кластеризация

Если в задачах кластеризации, которые мы рассмотрели выше требовалось разбить множество объектов на заданное число непересекающихся групп, то в задаче иерархической кластеризации (таксономии) *иерархической кластеризации*, или *таксономии*, необходимо найти иерархическое представление данных, такое, что кластеры на каждом уровне получаются объединением кластеров на более низком уровне. Таким образом, речь идет о кластерах кластеров. Понятно, что можно сказать по-другому: кластеры на более низком уровне получаются дроблением кластеров на более высоком уровне. В вершине этой классификации мы имеет один кластер, включающий все объекты. На низшем уровне мы имеем N кластеров, каждый из которых включает один объект. Очевидно, что такие иерархические структуры удобно представлять графически в виде деревьев (*дендрограмм*).

Подобные иерархические структуры встречаются в различных областях. Номенклатура живых существ, библиографические классификаторы, различные системы научной классификации и т. п. — все они являются примерами иерархической кластеризации (таксономии).

Для задачи (б) с разбиением экспериментов (клеток) на группы в зависимости их влияния на уровень экспрессии генов такое дерево[18] (одно из возможных) представлено на рис. 18. Оно получено агломеративным усредняющим алгоритмом, который кратко буден описан ниже. Иерархия составлена только на основе анализа уровня экспрессии генов и не учитывала информацию о заболевании. Однако из дендрограммы видно, что схожие клетки отнесены в близкие кластеры.

Алгоритмам иерархической кластеризации не нужно на вход подавать количество кластеров. Имея дендрограмму исследователь может сам обрезать ее на нужном уровне, получив некоторое количество кластеров.

Рассмотрим некоторые алгоритмы иерархической кластеризации. На вход алгоритма поступает группа из N объектов, для каждых двух из которых мы умеем измерять различие (расстояние). В примере с разбиением клеток на группы различие между i-й клеткой и i'-й можно мерить, например, по формуле

$$d_{ii'} = |x_{i1} - x_{i'1}| + |x_{i2} - x_{i'2}| + \ldots + |x_{ip} - x_{i'p}|,$$

где x_{ij} — уровень экспрессии j-го гена в i-й клетке, а p — число генов. В алгоритмах нам нужно будет уметь вычислять меру различия между двумя кластерами на основании попарных различий между объектами в этих кластерах. Пусть R, S — два кластера. Вот некоторые возможные способы определить различие d_{RS} между этими кластерами:

- по минимальному различию: в качестве d_{RS} выбирается минимальное различие между каждым объектом в R и каждым объектов в S;
- по максимальному различию: в качестве d_{RS} выбирается максимальное различие между каждым объектом в R и каждым объектов в S;

[18] То, что корень дерева традиционно располагается вверху, а листья внизу, не должно нас смущать

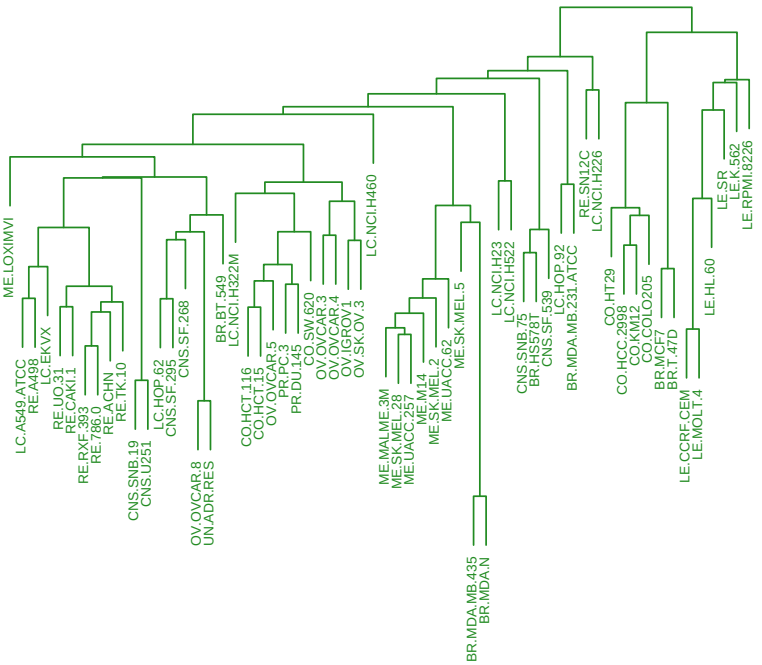

Рис. 18. Таксономия 60 клеток на основе анализа уровня экспресии их генов. Иерархия составлена агломеративным усредняющим методом.

- по усредненному различию: в качестве d_{RS} выбирается среднее значение расстояния между каждым объектом в R и каждым объектов в S.

Агломеративные методы, или методы «*снизу вверх*», строят дерево в направлении от листьев к корню. Все они начинают работу с N кластеров, каждый из которых содержит один объект. Найдем два самых близких друг к другу кластера и объединим их в один кластер. В полученном множестве кластеров снова найдем два ближайших друг к другу и объединим их в один кластер и т. д. пока не останется один кластер, включающий все объекты. Процесс заканчивается за $N - 1$ этапов.

Использование разных функций d_{RS} приводит к различным уточнениям данного метода. Используя в качестве d_{RS} мер различия по минимальному, максимальному и усредненному различиям, мы соответсвенно получаем так называемые алгоритм «простой связи» (single linkage), алгоритм «полной связи» (complete linkage), и «усредняющий» агломеративный алгоритм (group average). Первый из них обычно приводит к сильно несбалансированным деревьям, длинным и плохо связанным кластерам. Второй приводит к сбалансированным деревьям и «компактным» кластерам, но, с другой стороны, даже весьма близкие объекты могут оказаться в далеких друг от друга (по иерархии) кластерах. Третий метод занимает промежуточное положение между ними.

Структура деревьев, полученных в результате применения этих алгоритмов к

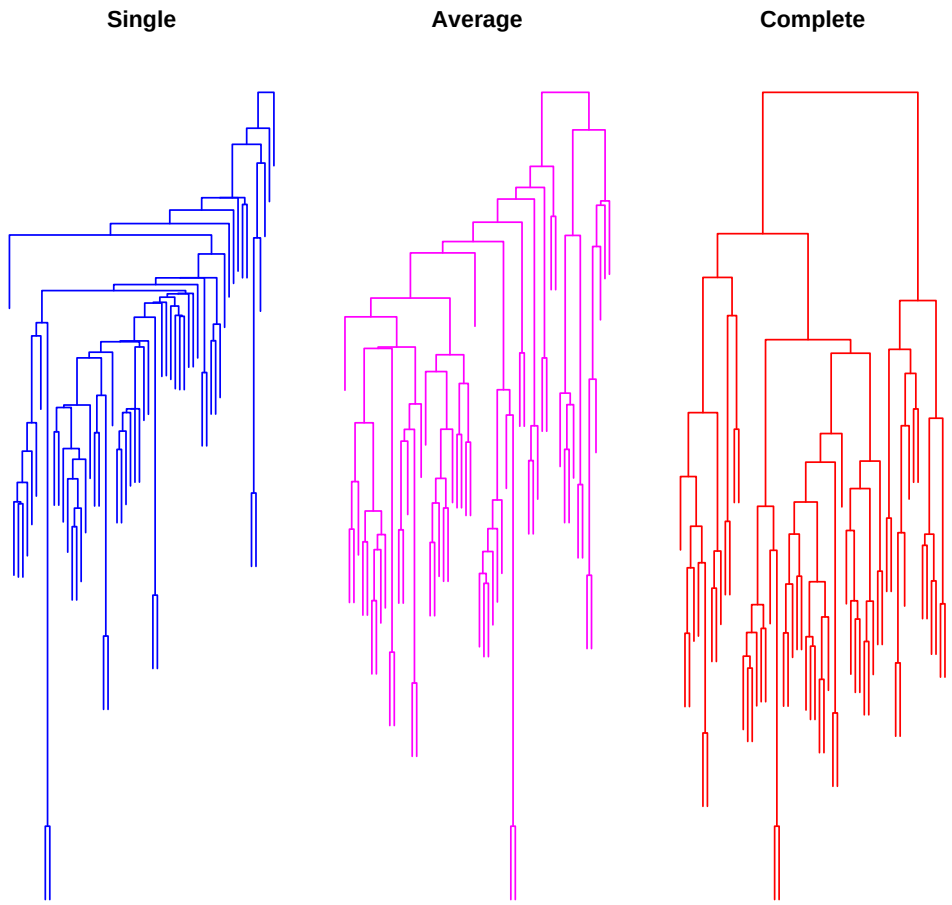

Рис. 19. Дендрограммы, полученные в результате применения алгоритма «простой связи», «усредняющего» агломеративного алгоритм и алгоритма «полной связи» к данным с биочипов.

данным с биочипов, приведена на рис. 19.

Наряду с агломеративными методами существуют *дивизивные*, или *разделяющие*, *методы* (методы «*сверху вниз*»). В них дерево строят в направлении от корня к листьям. На первом шаге ко множеству объектов можно применить какой-либо алгоритм (центров тяжести, медиан и т. п.), разбивающий это множество на два кластера. Затем разбивается каждый из полученных кластеров и т. д.

4.5. Списки Сводеша и таксономия языков

Список Сводеша (Swadesh) — список из 207 слов языка, заимствование которых из других языков (на поздних этапах) мало вероятно (местоимения, числительные 1–5, глаголы, обозначающие простые действия и т. п.) Вот фрагменты

этого списка для разных языков[19].

№	Русский	Английский	Немецкий	Итальянский	Французский	Чешский
1	я	I	ich	io	je	já
2	ты	you	du	tu	tu	ty
3	он	he	er	lui	il	on
4	мы	we	wir	noi	nous	my
5	вы	you	ihr	voi	vous	vy
6	они	they	sie	loro	ils	oni
7	этот	this	dieses	questo	ceci	tento
8	тот	that	jenes	quello	cela	tamten
9	здесь	here	hier	qui	ici	zde
10	там	there	dort	lá	lá	tam
11	кто	who	wer	chi	qui	kdo
12	что	what	was	che	quoi	co
13	где	where	wo	dove	où	kde
14	когда	when	wann	quando	quand	kdy
15	как	how	wie	come	comment	jak
16	не	not	nicht	non	ne. . . pas	ne
. . . . .						
205	если	if	wenn	se	si	jestlize
206	потому что	because	weil	perché	parce que	protoze
207	имя	name	Name	nome	nom	jméno

На основе анализа списков Сводеша для разных языков можно установить степень их родства и выделить группы родственных языков — это задача кластерного анализа. Более того, на основе анализа списка Сводеша для двух родственных языков можно приблизительно установить время их появляения из единого пра-языка.

Здесь объектами являются языки (или списки Сводеша), а их свойствами — элементы списка. Различие (расстояние) между списками Сводеша для двух языков определяется как количество пар соответствующих друг другу слов, не имеющих генетического родства, т. е. не родственных друг другу. Для нашего исследования мы упростили этот способ определения расстояния и вместо того, чтобы считать количество пар неродственных друг другу слов, нашли количество пар соответствующих друг другу слов, начинающихся с разных букв[20].

На рис. 20 графически представлена матрица различий между некоторыми языками[21]. Желтый цвет соответствует большим различиям, красный — малым. Мы специально переставили строки и столбцы, чтобы можно было обнаружить иерархическую структуру. Заметны кластеры, соответствующие языкам германской, романской, словянской и финно-угорской группам, а также кластер, содержащий два языка эльфов.

[19]Данные для примеров этого раздела взяты с сайта проекта Wiktionary `http://en.wiktionary.org/wiki/Wiktionary:Swadesh_list`.

[20]Очевидно, что этот способ, предложенный Д. Пойа, может привести к тому, что родственные слова мы будем считать неродственными (например, итальянское *io* и французское *je*) и наоборот, но понятно, что это не должно существенно сказаться на окончательной таксономии.

[21] Quenya и Sindarin — сконструированные Дж. Толкиеном языки эльфов.

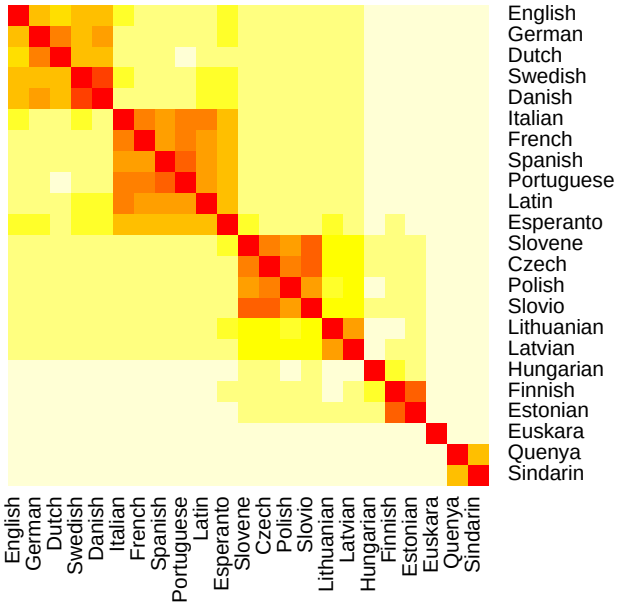

Рис. 20. Матрица сходства между некоторыми языками, построенная на основе списков Сводеша.

На рис. 21 приведено дерево иерархической кластеризации построенное усредняющим агломеративным методом. Несмотря на упрощенность нашего подхода, это дерево вполне соответсвует принятой таксономии языков.

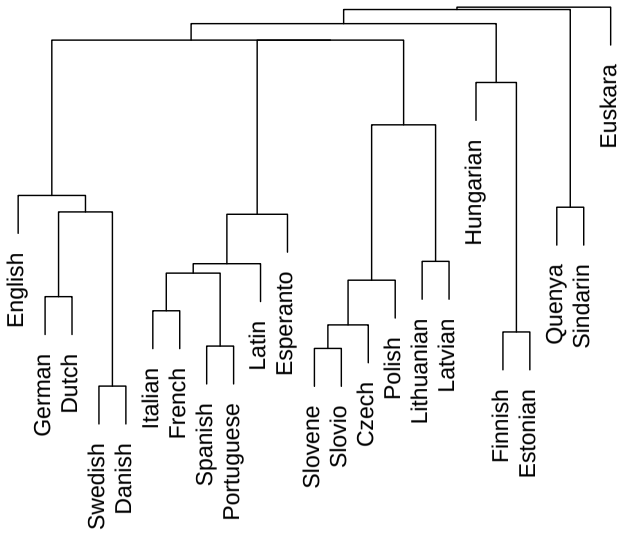

Рис. 21. Дерево иерархической кластеризации для 23 языков, построенное на основе списков Сводеша.