Открытый международный университет развития человека «Україна»

Институт компьютерных технологий

Кафедра інформаційних технологій та програмування

# РЕФЕРАТ

по дисциплине «Основы веб-программирования»

на тему: «Машинное обучение: базовые понятия и задачи»

Студентки 3-го курса

группы ПІ 31-14

специальность:

«Программная инженерия»

Любарской А.Е.

Киев- 2016

**Общие сведения о машинном обучении.**

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, невозможным, 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 вероятности, термин «машинное обучение» (Machine Learning) впервые ввёл в употребление А.Самуэль в 1959 году. В своей работе «Исследование в области машинного обучения на примере игры в шашки» он неформально определили понятие машинного обучения:

«Машинное обучение – процесс, в результате которого машина(компьютер) способна показывать поведение, которое в нее не было явно заложено»

*A.L.Samuel* Some Studies in Machine Learning Using the Game of Checkers

Самуэль описывает компьютерную программу, умеющую играть в шашки и умеющую обучаться в результате 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 даст более строгое определение термину «машинное обучение»:

«Говорят, что компьютерная программа обучается на основе опыта Е по отношению к некоторому классу задач Т и меры качества Р, если качество решения задач из Т, измеренное на основе Р, улучшается с приобретение опыта Е»

*T.M.Mitchell* Machine Learning

Как же научить машины обучаться? Существуют разные подходы. В 1958 году Ф.Розенблатт (*F.Rosenblatt* The Perceptron. A Probablistic Model for Information Storage and Organisation in the Brain) 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 интеллектуальных задач – нужно только увеличить число нейронов и посмотреть получше, как устроен человеческий мозг. Возможности современной техники позволяют значительно увеличить количество нейронов, но не в количестве дело. Вапник (*V.Vapnik* The Nature of Statistical Learning Theory) отмечает:

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

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, ещё далеко до конструирования *универсального* решателя интеллектуальных задач, но *конкретные* сложные задачи мы можем научить решать компьютер уже сейчас. При этом, разумеется, находят своё применение классические методы (регрессия и метод наименьших квадратов, статистическая оценка параметров функции распределения, дискриминантный анализ и др.), но, конечно, появляются и бурно развивающиеся 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 её 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 предсказывало бы выход. Признаки объекта называются также *входами*. Таким образом, по входам требуется научиться предсказывать выход.

Признаки и выходы бывают *количественные* (например, цена, температура) и *качественные,* или *номинальные* (например, пол, название заболевания, отсутствие/присутствие конкретного симптома). 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.

Практическое применение машинного обучения:

\*Распознавание речи.

\*Распознавание жестов.

\*Распознавание рукописного ввода.

\*Распознавание образов.

\*Техническая диагностика.

\*Медицинская диагностика.

\*Прогнозирование временных рядов.

\*Биоинформатика.

\*Обнаружение мошенничества.

\*Обнаружение спама.

\*Категоризация документов.

\*Биржевой технический анализ.

\*Финансовый надзор.

\*Кредитный скоринг.

\*Предсказание ухода клиентов.

\*Хемоинформатика.

\*Обучение ранжированию в информационном поиске.

**Регрессия.**

**«Регрессия к середине» Ф. Гальтона.**

В 1885 году в статье «Регрессия к середине в наследовании роста» (*F. Galton* Regression towards Mediocrity in Hereditery Stature) 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 называл *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

*(x1, y1), (x2, y2), …, (xn, yn), (1)*

Где *xi* – средний рост двух родителей, а *yi* – рост их взрослого ребёнка, N = 928. Для учёта того, что мужчины в среднем выше женщин, использовались поправочные коэффициенты. *Диаграмма рассеяния* для данных Гальтона представлена на рис.1. У Гальтона данные приведены с 1 знаком после запятой. В этом случае многие точки совпали бы друг с другом. Чтобы они выглядели как «облако», мы добавили к координатам каждой точки малое случайное возмущение.

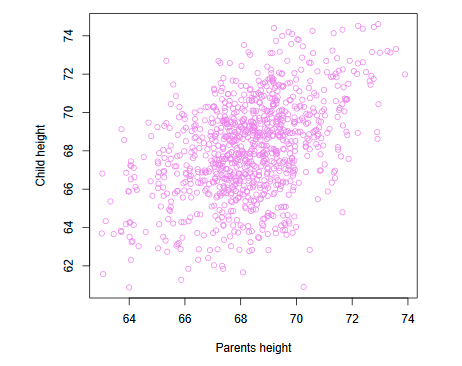


Рис.1. Диаграмма рассеяния для данных из исследования Ф. Гальтона. По горизонтальной оси откладывается рост родителей *х*, по вертикальной – рост детей *у*.

Попробуем сами проанализировать эти данные и получить аналогичный результат.

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

*y = β0 + β1x1 (2)*

где *β0, β1* – пока не известные коэффициенты. Таким образом,

*yi* ≈ *β0 + β1 x1 (i = 1,2, …., N).*

Если бы *β0, β1* были известны, то по формуле (2) мы могли бы предсказывать значение *у* по заданному *х*. Как правило, мы будем совершать некоторую ошибку, но тенденция будет угадана правильно. Рассмотрим *остаточную сумму квадратов*, равную

*R(β0, β1) = .*

Здесь *yi* *-* *β0 - β1xi* равно отклонению предсказанного значения *β0 - β1xi* от истинного *yi* и, таким образом, *R(β0, β1)* характеризуют общую ошибку на имеющихся данных.

В качестве *β0, β1* выберем значения, на которых достигается минимум функции *R(β0, β1)*. Находим частные производные:

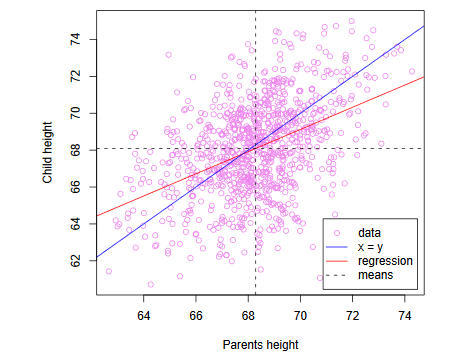
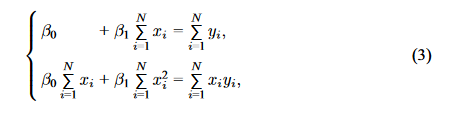


Рис.2. Зависимость роста взрослого ребёнка от роста родителей в исследовании Ф. Гальтона. По горизонтальной оси откладывается рост родителей *х*. По вертикальной – рост детей в дюймах, *у*. Синяя прямая имеет уравнение *у = х*. Красная прямая построена по методу наименьших квадратов по эмпирическим данным. 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*

**

относительнонеизвестных *β0, β1*. Система имеет решение *β0 =* 24*, β1 =* 0.65*.* Таким образом, восстановленная зависимость имеет вид

*у =* 24 *+* 0.65 *х. (4)*

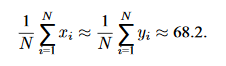
График восстановленной зависимости приведен на рис.2.

Метод, с помощью которого мы нашли *β0, β1*, 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:

*y ≈* 68.2 + 0.65 \* (*x* – 68.2) *(5)*

Здесь 68.2 приближённо равно среднему значению для *xi* и для *yi :*

**

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 неё».

В терминологии машинного обучения последовательность (1) – это обучающая выборка, *xi* – объекты, а *yi* – 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 *β0, β1* – *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 = движение назад) общее правило. Такой метод в корне отличается от традиционного для того времени подхода, когда данные только подтверждали закон.

**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, удалённость до основных магистралей, наличие инфраструктуры, экологическую обстановку в районе. Как, пользуясь всей этой информацией, определить цену вашего жилья?

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

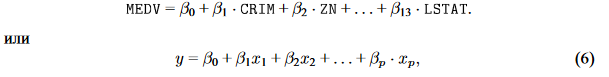
Для примера рассмотрим Boston Housing Data (UCI Repository of Machine Learning Databases). База содержит информацию о загородных домах близ Бостона. Данные были собраны в 1970-х годах. Информация агрегирована: территория поделена на участки и дома, стоящие на одном участке, собраны в группы. Таким образом, объектами являются сами эти группы. Их общее количество – 506. В качестве признаков рассматриваются:

1. CRIM – уровень преступности на душу населения.
2. ZN – процесс земли, застроенной жилыми домами (только для участков площадью свыше 25000 кв. футов).
3. INDUS – процент деловой застройки.
4. CHAS – 1, если участок граничит с рекой; 0 в противном случае (бинарный признак).
5. NOX – концентрация оксида азота, делённая на .
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 удалённости до радиальных магистралей.
10. TAX – величина налога $ 10000.
11. PTRATIO – количество учащихся, приходящихся на одного учителя (по городу).
12. B = 1000 ( АА – 0.63)², где АА – доля афро-американцев.
13. LSTAT – процент жителей с низким социальным статусом.

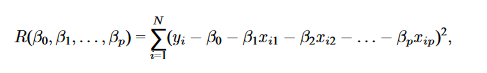
Признак CHAS – бинарный, остальные – количественные. Выходом является переменная MEDV, равная медианному значению цены строения (по всем домам участка) в $ 1000.

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

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



Где *р =* 13, а переменные MEDV, CRIM, ZN, … , LSTAT обозначены соответственно через *у, х1, х2, … , хр*. 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



Где *уі, хі1, хі2, … , хір* – значения переменных *у, х1, х2, … , хр* соответственно для *і-го* объекта из обучающей выборки (*і-го* участка в базе). Взяв частные производные по параметрам *β0, β1, …, βр*, и приравняв их к нулю, приходим к системе уравнений, аналогичной (3), из которой определяем значения параметров

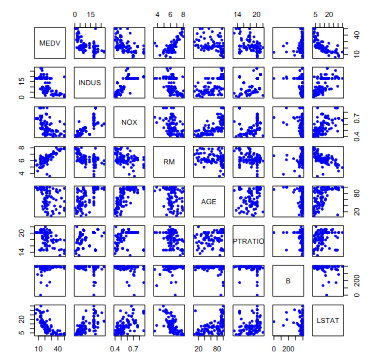


Рис.3. Диаграммы рассеяния для каждой пары переменных MEDV, INDUS, NOX, RM, AGE, PTRATIO, B в задаче о предсказании стоимости дома. Значение переменной MEDV нужно научиться предсказывать по значениям остальных переменных.

**Классификация.**

**Как предсказать уровень гормона?**

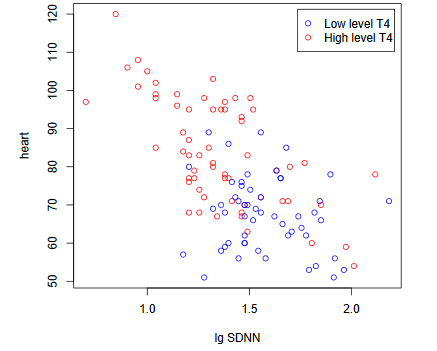
****

Рис.4. Диаграмма, представляющая данные о группе лиц с заболеваниями 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 Т4.

На рис.4 представлены данные о 114 лицах с заболеванием щитовидной железы (данные представлены врачом больницы 13 Нижнего Новгорода М.Н. Будкиной). На момент проведения обследования у 61 из них был повышенный уровень гормона Т4 (гиперфункция щитовидной железы), у 53 пациентов уровень этого гормона был в норме. Для каждого пациента известны следующие показатели: ЧСС – частота сердечных сокращений и SDNN – стандартное отклонение длительности интервалов между синусовыми сокращениями RR. Измерение ЧСС и SDNN проще и дешевле установления уровня Т4, поэтому возникает вопрос, можно ли научиться предсказывать (допуская небольшие ошибки) уровень свободного Т4 по ЧСС и SDNN.

На рис.5 изображены те же точки и, кроме того, прямая с уравнением *16\* lg SDNN – ЧСС + 50 = 0 , (7)*

разбивающая плоскость на две области. Мы видим, что она достаточно удачно разбивает множество точек, так, что по одну её 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 соответствующая его показателям точка и на основании этого предсказать, какой у него уровень Т4. На языке алгебры это означает следующее. Для данных SDNN и ЧСС необходимо вычислить функцию

*f (SDNN, ЧСС) = 16 lg SDNN – ЧСС + 50.*

*Решающее правило* заключается в следующем. Если *f (SDNN, ЧСС)>* 0, то у пациента нормальный уровень Т4, а если *f (SDNN, ЧСС)<0,* то уровень Т4 повышенный. Данное решающее правило совершает ошибку в 23% случаев.

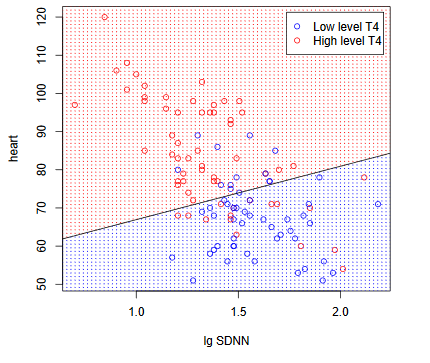


Рис.5. Прямая разделяет плоскость на две области и тем самым определяет решающее правило.

Данная задача – пример *классификации*. По обучающей выборке

*(x1, y1), (x2, y2), …, (xn, yn),*

Где *хі –* пациент из базы данных, а *уі –* класс, к которому он принадлежит (например, *уі=0* означает, что уровень Т4 у него в норме, *уі=1* – повышенный уровень) мы научимся предсказывать, к какому классу будут принадлежать новые пациенты, то есть построим решающее правило. Таким образом, задача классификации похожа на задачу восстановления регрессии и отличается от неё 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, то есть пространство признаков было двумерным. Нам удалось найти прямую, которая задаёт достаточно хорошее решающее правило. В трёхмерном пространстве аналогом прямой является плоскость, а в пространствах большей размерности – гиперплоскость. Достаточно часто хорошее решающее правило удаётся 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.

**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 равно 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, то есть, 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.

**Машина опорных векторов.**

Познакомимся с основами знаменитого метода построения решающего правила, носящего название «машина опорных векторов». Этот метод был предложен Вапником и Червоненкисом в 1974 г. (*В.Н. Вапник, А.Я. Червоненкис* Теория распознавания образов. Название «машина опорных векторов» метод приобрёл позднее.) 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 неё 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 коснётся какой-либо точки из обучающей выборки. Область, которую при этом заметёт прямая, окрашена на рис.6 в жёлтый цвет. Назовём 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. Назовём эту прямую *оптимальной*.

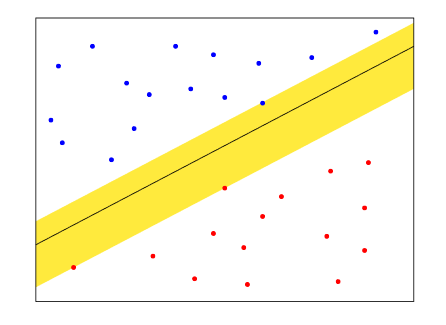


Рис.6. Прямая точно разделяет множество на два заданных класса. Нейтральная полоса закрашена жёлтым цветом.

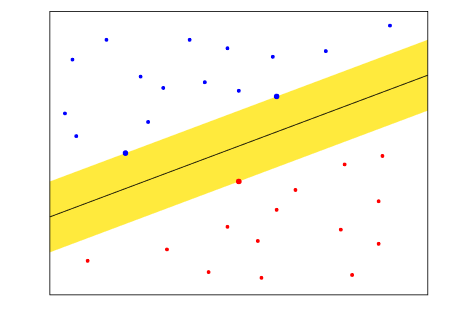


Рис.7. Оптимальная разделяющая прямая. Нейтральная полоса имеет максимальную ширину, и разделяющая прямая лежит в точности посередине её.

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

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

Что делать, если точки нельзя разделить гиперплоскостью на два заданных класса? В этом случае решают несколько изменённую задачу. В её 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.

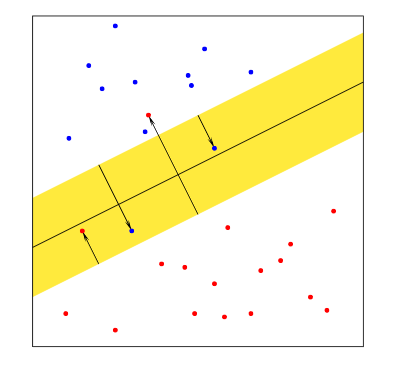


Рис.8. Точки нельзя разделить прямой на два заданных класса. Точки, которые попали на нейтральную полосу или зашли на «чужую территорию», отмечены стрелками.

**Распознавание рукописных символов.**

Рассмотрим задачу распознавания символов. Выборка optidigit (UCI Repository of Machine Learning Databases) содержит информацию о 1934 изображениях рукописных цифр, некоторые из них представлены на рис.9.

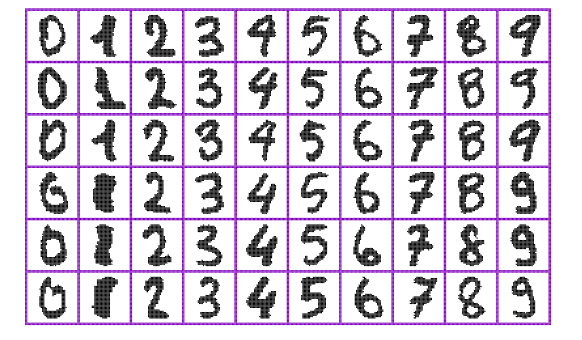
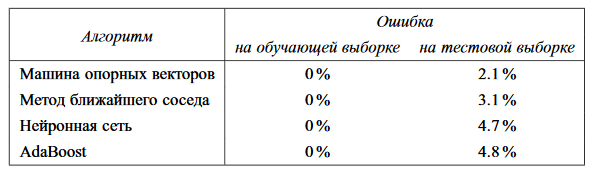


Рис.9. Некоторые объекты из обучающей выборки в задаче optidigit распознавания рукописных цифр.

Каждое изображение закодировано вектором длины 1024, составленным 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 нём представлено (то есть известен класс). Получили задачу классификации, при этом объектами являются изображения, входами – бинарные признаки *х1, х2, … , х1024*, 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, то есть настройки параметров классификаторов. На второй выборке (контрольной) путём вычисления ошибки оценивалась его обобщающая способность.

Ошибки на обучающей и тестовой выборках приведены в следующей таблице.



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.10. Изображения, неправильно 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.

**Кластеризация.**

**Извержение гейзера.**

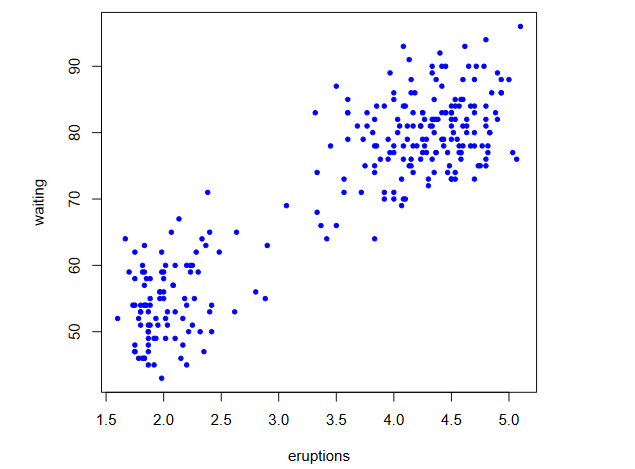
****

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

На рис.11 представлены данные о времени между извержениями и длительностью извержения Old Faithful geyser in Yellowstone National Park, Wyoming, USA (*A. Azzalini, A.W. Bowman* A look at some data on the Old Faithful geyser). 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 неизвестен, 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.

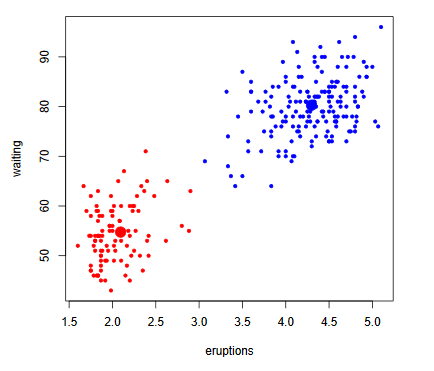


Рис.12. Данные разбиты на два класса. Жирными точками отмечены центры тяжести кластеров.

Рассмотрим *метод центров тяжести* для решения задачи кластеризации. Пусть множество объектов требуется разбить на К кластеров. Вначале произвольным образом разобьём объекты на К классов. В каждой группе найдём центр тяжести и проведём 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.

**Иерархическая кластеризация.**

Если в задачах кластеризации, которые были рассмотрены выше, требовалось разбить множество объектов на заданное число непересекающихся групп, то в *задаче иерархической кластеризации (таксономии)* необходимо найти иерархическое представление данных, такое, что кластеры на каждом уровне получаются объединением кластеров на более низком уровне. Таким образом, речь идёт о кластерах кластеров. Понятно, что можно сказать по-другому: кластеры на более низком уровне получаются дробление кластеров на более высоком уровне. В вершине этой классификации мы имеем 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 – все они являются примерами иерархической кластеризации (таксономии).

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

**Экспрессия генов и иерархическая кластеризация.**

*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 определённых 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 биочип к генам прикрепляются молекулы флуоресцентного вещества. Светимость точки пропорциональна уровню экспрессии соответствующего гена. Если точка не светится, то заданного гена в исследуемом материале нет. Информацию можно представить в виде длинного числового вектора. Каждая его компонента соответствует определённому гену. Значение, хранящееся в этой компоненте (то есть 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 соответствуют увеличению этого уровня по сравнению с контрольным. Отрицательные значения – уменьшению. Перед помещением исследуемый и контрольный генетический материал необходимым образом обрабатываются. В частности, к материалам прикрепляются молекулы двух разных флуоресцентных веществ, что приводит к окрашиванию точек матрицы в оттенки двух цветов. Например, если используются Су3 (зелёный) и Су5 (красный), то получим цвета от ярко-зелёного до ярко-красного. Зелёный цвет означает, что выше уровень экспрессии контрольного материала, красный – что выше уровень экспрессии у исследуемого материала, серый – что уровень экспрессии одинаков. Эту информацию можно закодировать в виде числового вектора. Его положительные, отрицательные и нулевые значения означают, что уровень экспрессии у исследуемого материала соответственно выше, ниже контрольного уровня или равен ему. Например, если исследуемый материал получен из исходного введением какого-либо лекарства, то положительные значения соответствуют повышению уровня экспрессии, отрицательные – уменьшению, нулевые означают, что уровень не изменился.

Условное изображение некоторого биочипа приведено на рис.13.

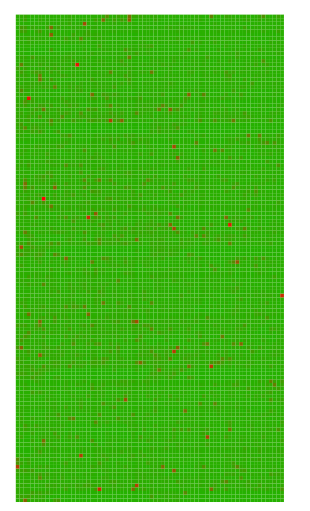


Рис.13. Условное изображение биочипа. Каждая точка на рисунке соответствует определённому гену. Всего анализируется 9504 гена.

Анализируется генетический материал из поражённых болезнью Паркинсона клеток мозга мыши. Биочип выглядит как матрица размером 132\*72. Каждая точка на рисунке соответствует определённому гену. Таким образом, всего анализируется 132\*72 = 9504 гена. Красный цвет показывает, что уровень экспрессии больной клетки выше нормы, а зелёный – ниже нормы. В частности, максимальный уровень экспрессии (по сравнению с контрольным уровнем) показывает ген Gapd (ярко-красная точка), а минимальный – ген АА395996 (зелёная 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 соответствуют разным генам, а столбцы – разным экспериментам (разным клеткам). Рассмотрим следующие задачи:

А) Разбить гены на группы в зависимости от влияния на них экспериментов. Гены, реагирующие «почти одинаковым» образом в «большом числе» экспериментов, 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». Имеется информация о 1375 генах в 60 различных клетках. Графически данные представлены в виде матрицы на рис.14.



Рис.14. 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.

Для задачи Б) с разбиением экспериментов (клеток) на группы в зависимости от их влияния на уровень экспрессии генов, дендрограмма (одно из возможных деревьев) представлено на рис.15.

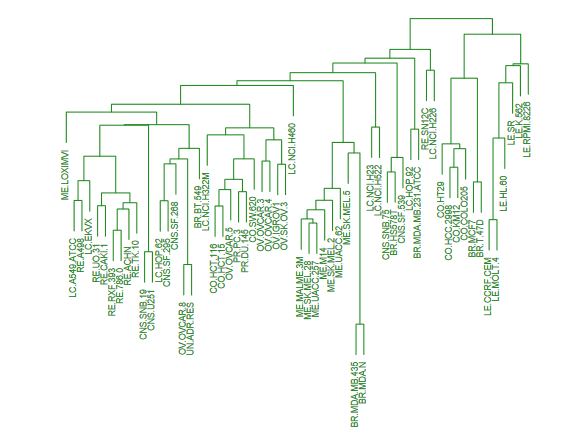


Рис.15. Таксономия 60 клеток на основе анализа уровня экспрессии их генов. Иерархия составлена агломеративным усредняющим методом.

Оно получено агломеративным усредняющим алгоритмом, который кратко будет 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.

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 *і*-ой клеткой и *і’*-й можно мерить, например, по формуле

F:\WEB\реферат\формуладерево.png

Где *хij* – уровень экспрессии *j-*го гена в *і-*й клетке, а *p* – число генов. В алгоритме 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 *dR S* между этими кластерами:

-по минимальному различию: в качестве *dR S* выбирается минимальное различие между объектом в R и каждым объектом в S.

-по максимальному различию: в качестве *dRS* выбирается максимальное различие между объектом в R и каждым объектом в S.

-по усреднённому различию: в качестве *dRS* выбирается среднее значение расстояния между каждым объектом в R и каждым объектом 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. Найдём два самых близких друг к другу кластера и объединим их в один кластер. В полученном множестве кластеров снова найдём два ближайших друг к другу и объединим их в один кластер, и так далее, пока не останется один кластер, включающий все объекты. Процесс заканчивается за N – 1 этапов.

Использование разных функций *dRS* приводит к различным уточнениям данного метода. Используя в качестве *dRS мер* различия по минимальному, максимальному и усреднённому различиям, мы соответственно получаем так называемый алгоритм «простой связи» (single linkage), алгоритм «простой 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 далёких друг от друга (по иерархии) кластерах. Третий метод занимает промежуточное положение между ними.

Структура деревьев, полученных в результате применения этих алгоритмов к данным с биочипов, показана на рис.16.

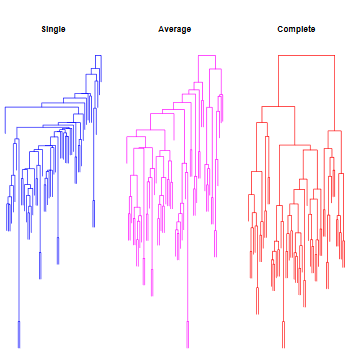


Рис.16. Дендрограммы, полученные в результате применения алгоритма «простой связи», «усредняющего» агломеративного алгоритма и алгоритма «полной связи» к данным с биочипов.

Наряду с агломеративными методами, существуют *дивизивные (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, меридиан), разбивающий это множество на два кластера. Затем разбивается каждый из полученных кластеров, и так далее.