Вычислительные методы обучения по прецедентам. Введение

К. В. Воронцов

19 января 2009 г.

Материал находится в стадии разработки, может содержать ошибки и неточности. Автор будет благодарен за любые замечания и предложения, направленные по адресу `vokov@forecsys.ru`, либо высказанные в обсуждении страницы «Машинное обучение (курс лекций, К.В.Воронцов)» вики-ресурса `www.MachineLearning.ru`.

Содержание

1 Введение: задачи обучения по прецедентам

В этой вводной главе даются базовые понятия и обозначения, которые будут использоваться на протяжении всего курса. Приводится общая постановка задачи и намечаются основные проблемы, с которыми придётся сталкиваться в дальнейшем.

В качестве иллюстрации перечисляется большое количество актуальных прикладных задач, для решения которых могут применяться методы и алгоритмы, рассматриваемые в последующих главах.

Введение завершается замечаниями обще-методологического характера, которые по традиции помещаются во введениях, хотя более уместно было бы поместить их в заключении. Они будут особенно полезны тем, кто уже приобрёл минимальный собственный опыт решения прикладных задач и готов воспринять обобщение чужого опыта в форме сухих рекомендаций.

§1.1 Основные понятия и определения

Пусть имеются множество *объектов* X, множество *ответов* Y, и существует *целевая функция* (target function) $y^*\colon X \to Y$, значения которой $y_i = y^*(x_i)$ известны только на конечном подмножестве объектов $\{x_1, \ldots, x_\ell\} \subset X$. Пары «объект–ответ» (x_i, y_i) называются *прецедентами*. Совокупность пар $X^\ell = (x_i, y_i)_{i=1}^\ell$ называется *обучающей выборкой* (training sample).

Задача *обучения по прецедентам* заключается в том, чтобы восстановить функциональную зависимость между объектами и ответами, то есть построить отображение $a\colon X \to Y$, удовлетворяющее следующей совокупности требований:

- Отображение a должно допускать эффективную компьютерную реализацию. По этой причине будем называть его *алгоритмом*.
- Алгоритм $a(x)$ должен воспроизводить на объектах выборки заданные ответы: $a(x_i) = y_i$, $i = 1, \ldots, \ell$. Равенство здесь может пониматься как точное или как приближённое, в зависимости от конкретной задачи.
- Алгоритм a должен обладать *обобщающей способностью* (generalization ability), то есть достаточно точно приближать целевую функцию $y^*(x)$ не только на объектах обучающей выборки, но и на всём множестве X.
- На алгоритм $a(x)$ могут накладываться разного рода *априорные ограничения*, такие, как непрерывность, гладкость, монотонность, устойчивость, и т. д. В некоторых случаях задаётся функциональный вид (*модель*) алгоритма $a(x)$.

Пока мы описали постановку задачи довольно неформально. Далее мы уточним понятия «зависимость между объектами и ответами», «точность ответов», «обобщающая способность», «модель алгоритма».

1.1.1 Разновидности задач обучения по прецедентам

В зависимости от природы множества Y задачи обучения по прецедентам делятся на следующие типы:

- $Y = \{1, \ldots, M\}$ — задача *классификации* (classification) на M непересекающихся классов. В этом случае всё множество объектов X разбивается на классы $K_y = \{x \in X : y^*(x) = y\}$, и алгоритм $a(x)$ должен давать ответ на вопрос «какому классу принадлежит x?». В некоторых приложениях классы называют *образами* и говорят о задаче *распознавания образов* (pattern recognition).

- $Y = \{0, 1\}^M$ — задача *классификации на M пересекающихся классов*. В простейшем случае эта задача сводится к решению M независимых задач классификации с двумя непересекающимися классами.

- $Y = \mathbb{R}$ — задача *восстановления регрессии* (regression estimation).

- Задача *прогнозирования* (forecasting) является частным случаем классификации или восстановления регрессии, когда X — описание прошлого поведения объекта, Y — описание некоторых характеристик его будущего поведения. Более строгая постановка задачи будет дана в главе **??**.

Замечание 1.1. При рассмотрении алгоритмов классификации часто ограничиваются случаем двух классов, более удобным для теоретического анализа. Существует несколько стандартных способов свести задачу с M классами к задаче с двумя классами. Например, можно по очереди отделить каждый класс от остальных, решив M задач классификации с двумя классами K_y и $K_{\bar{y}} = \{x \in X \mid y^*(x) \neq y\}$.

1.1.2 Модель алгоритмов и метод обучения

Опр. 1.1. *Моделью алгоритмов называется параметрическое семейство отображений A, из которого выбирается искомый алгоритм $a(x)$:*

$$A = \{\varphi(x, \gamma) \mid \gamma \in \Gamma\},$$

где $\varphi : X \times \Gamma \to Y$ — некоторая фиксированная функция, Γ — множество допустимых значений параметра γ, называемое пространством параметров или пространством поиска (search space).

Процесс подбора параметров модели по обучающей выборке называют *настройкой* (fitting) или *обучением* (training, learning) алгоритма[1]. В результате настройки выбирается единственный алгоритм $a \in A$, который должен приближать целевую зависимость.

Опр. 1.2. *Методом обучения называется отображение $\mu : (X \times Y)^\ell \to A$, которое произвольной конечной выборке X^ℓ ставит в соответствие алгоритм $a : X \to Y$. Говорят также, что метод μ строит алгоритм a по выборке X^ℓ. Метод обучения, как и сам алгоритм a, должен допускать эффективную программную реализацию.*

Итак, в задачах обучения по прецедентам чётко различаются два этапа:

- на этапе *обучения* метод μ по выборке X^ℓ строит алгоритм $a = \mu(X^\ell)$;

[1] Английская терминология тонко различает, что алгоритм является обучаемым, учеником (learning machine), а выборка данных — обучающей, учителем (training sample).

- на этапе *применения* алгоритму a подаются на вход новые объекты x, вообще говоря, отличные от обучающих, для получения ответов $y = a(x)$.

Этап обучения наиболее сложен. Как правило, он сводится к поиску параметров модели, доставляющих оптимальное значение заданному функционалу качества.

1.1.3 Функционал качества

Опр. 1.3. *Функция потерь (loss function) — это неотрицательная функция $\mathscr{L}(a, x)$, характеризующая величину ошибки алгоритма a на объекте x. Если $\mathscr{L}(a, x) = 0$, то ответ $a(x)$ называется корректным.*

Опр. 1.4. *Функционал качества алгоритма a на выборке X^ℓ:*

$$Q(a, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathscr{L}(a, x_i). \tag{1.1}$$

Функционал Q называют также функционалом *средних потерь* или *эмпирическим риском* [1], так как он вычисляется по *эмпирическим данным* $(x_i, y_i)_{i=1}^\ell$.

Функция потерь, принимающая только значения 0 и 1, называется *бинарной*. В этом случае $\mathscr{L}(a, x) = 1$ означает, что алгоритм a допускает ошибку на объекте x, а функционал Q называется *частотой ошибок* алгоритма a на выборке X^ℓ.

Пример 1.1. Наиболее употребительные функции потерь при $Y \subseteq \mathbb{R}$:

- $\mathscr{L}(a, x) = [a(x) \neq y^*(x)]$ — индикатор несовпадения с правильным ответом (обычно применяется в задачах классификации);
- $\mathscr{L}(a, x) = \big[|a(x) - y^*(x)| \geqslant \varepsilon\big]$ — индикатор существенного отклонения от правильного ответа, где ε — заданный порог точности.
- $\mathscr{L}(a, x) = |a(x) - y^*(x)|$ — отклонение от правильного ответа; функционал Q называется *средней ошибкой* алгоритма a на выборке X^ℓ;
- $\mathscr{L}(a, x) = (a(x) - y^*(x))^2$ — квадратичная функция потерь; функционал Q называется *средней квадратичной ошибкой* алгоритма a на выборке X^ℓ.
- $\mathscr{L}_w(a, x) = w(x)\mathscr{L}(a, x)$ — взвешенная функция потерь, где $w(x)$ — неотрицательная *весовая функция*, характеризующая степень важности объекта x; $\mathscr{L}(a, x)$ — какая-либо невзвешенная функция потерь, например, одна из перечисленных выше.

Классический метод обучения, называемый *минимизацией эмпирического риска* (empirical risk minimization, ERM), заключается в том, чтобы найти в заданной модели A алгоритм a, доставляющий минимальное значение функционалу качества Q на заданной обучающей выборке X^ℓ:

$$\mu(X^\ell) = \arg\min_{a \in A} Q(a, X^\ell). \tag{1.2}$$

В следующем параграфе будет рассмотрен вероятностный подход к обучению, при котором возникает похожая оптимизационная задача.

1.1.4 Вероятностная постановка задачи обучения

До сих пор мы упускали из виду, что элементы множества X — это не реальные объекты, а лишь их описания, доступная информация об объектах. Полные описания практически никогда не бывают известны. Мы не умеем исчерпывающим образом охарактеризовать человека, геологический район, производственное предприятие или экономику страны. Поэтому одному и тому же описанию x могут соответствовать различные объекты, а, значит, и целое «облако ответов» $y^*(x)$. Для формализации этих соображений вводится вероятностная постановка задачи. Вместо существования неизвестной целевой функции $y^*(x)$ предполагается существование неизвестного вероятностного распределения на множестве $X \times Y$ с плотностью $p(x, y)$, из которого случайно и независимо выбираются ℓ наблюдений $X^\ell = (x_i, y_i)_{i=1}^\ell$. В математической статистике такие выборки называются *простыми* или *случайными одинаково распределёнными* (independent identically distributed, i.i.d.).

Вероятностная постановка задачи считается более общей [1, 20], так как функциональную зависимость $y^*(x)$ можно представить в виде вероятностного распределения $p(x, y) = p(x)p(y|x)$, положив $p(y|x) = \delta(y - y^*(x))$, где $\delta(z)$ — дельта-функция. Однако при этом приходится вводить дополнительную гипотезу о существовании на множестве X неизвестного распределения $p(x)$. Функциональная постановка задачи никак не связана с вероятностными предположениями, поэтому называть её частным случаем вероятностной не вполне корректно.

Адекватна ли гипотеза о существовании распределений $p(x)$ и $p(y|x)$ практическому опыту — вопрос скорее философский, и мы оставим его за рамками обсуждения. Многие исследователи соглашаются с этой гипотезой просто потому, что она позволяет привлечь удобный математический аппарат теории вероятностей. Мы также оставим без ответа вопрос о правомерности трактовать неопределённость, связанную с недостатком информации, как вероятностное распределение $p(y|x)$. Существуют и другие подходы, в частности, теория возможности Пытьева [10] и теоретико-множественный подход Трауба и Васильковского [12].

В данном курсе задача обучения по прецедентам будет рассматриваться только в функциональной или вероятностной постановке.

Принцип максимума правдоподобия. Плотность распределения простой выборки $p(X^\ell)$ равна произведению значений плотности $p(x, y)$ в отдельных наблюдениях:

$$p(X^\ell) = p\big((x_1, y_1), \ldots, (x_\ell, y_\ell)\big) = p(x_1, y_1) \times \cdots \times p(x_\ell, y_\ell).$$

Если подставить сюда вместо y_i ответы алгоритма $a(x_i)$, то получится *функция правдоподобия* (likelihood), оценивающая, насколько хорошо ответы алгоритма a согласуются с распределением $p(x, y)$:

$$L(a, X^\ell) = \prod_{i=1}^{\ell} p(x_i, a(x_i)).$$

Чем выше значение правдоподобия, тем лучше алгоритм a согласуется с распределением $p(x, y)$. Значит, нужно искать алгоритм a, доставляющий максимум функционалу $L(a, X^\ell)$. В математической статистике это называется *принципом максимума правдоподобия*. Его формальные обоснования можно найти в [6].

Вместо максимизации L удобно минимизировать функционал $-\log L$, который представляется в виде суммы:

$$-\ln L(a, X^\ell) = -\sum_{i=1}^{\ell} \ln p(x_i, a(x_i)).$$

Это выражение по форме совпадёт с функционалом эмпирического риска (1.1), если определить *вероятностную функцию потерь* $\mathscr{L}(a, x) = -\ell \ln p(x, a(x))$. Чем меньше вероятность пары объект–ответ $(x, a(x))$, тем выше величина потери $\mathscr{L}(a, x)$.

Существует любопытная взаимосвязь между вероятностной постановкой и функциональной. Сделаем дополнительное предположение, что ошибки $\varepsilon(x) = a(x) - y^*(x)$ имеют нормальное распределение $\mathcal{N}(\varepsilon; 0, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right)$ с нулевым средним и дисперсией σ^2. Тогда вероятностная функция потерь совпадёт с квадратичной с точностью до констант $c_0 = \ln(\sigma\sqrt{2\pi})$, $c_1 = \frac{1}{2\sigma^2}$:

$$-\ln p(x, a(x)) = -\ln \mathcal{N}\big(a(x) - y^*(x); 0, \sigma^2\big) = c_0 + c_1(a(x) - y^*(x))^2.$$

Аналогично устанавливается эквивалентность вероятностных и функциональных постановок практически для любых «разумных» функций потерь. Таким образом, максимизация правдоподобия — это та же минимизация эмпирического риска, причём вид функции потерь взаимно однозначно связан с априорным предположением о распределении ошибок.

В одних прикладных задачах предпочтительнее вероятностная постановка, в других — функциональная. Если предполагается, что величина ошибки складывается из многих случайных факторов, то можно считать, что её распределение нормально и пользоваться методом наименьших квадратов. Однако есть приложения, в которых гипотезы о распределении ошибок не достаточно хорошо обоснованы. Например, в эконометрике они зачастую лишены экономического смысла [9], зато функция потерь строится легко и выражает буквально потери в рублях. При этом она запросто может оказаться не квадратичной и даже не симметричной.

Отметим, что обе постановки, вероятностная и функциональная, сводятся в итоге к однотипным оптимизационным задачам.

1.1.5 Проблема переобучения и понятие обобщающей способности

Минимизацию эмпирического риска следует применять с известной долей осторожности, поскольку главным её недостатком является склонность к переобучению. Если алгоритм a доставляет минимум функционалу $Q(a, X^\ell)$ на заданной обучающей выборке X^ℓ, это ещё не гарантирует, что он будет хорошо приближать целевую зависимость на произвольной *контрольной выборке* $X^k = (x'_i, y'_i)_{i=1}^k$.

Когда качество работы алгоритма на новых объектах, не вошедших в состав обучения, оказывается существенно хуже, чем на обучающей выборке, говорят об эффекте *переобучения* (overtraining) или *переподгонки* (overfitting). При решении практических задач с этим явлением приходится сталкиваться очень часто.

Пример 1.2. Легко представить себе метод, который минимизирует эмпирический риск до нуля, но при этом абсолютно не способен обучаться. Получив обучающую выборку X^ℓ, он запоминает её и строит алгоритм, который сравнивает предъявляемый

объект x с обучающими объектами x_i из X^ℓ. В случае совпадения $x = x_i$ алгоритм выдаёт правильный ответ y_i. Иначе выдаётся произвольный ответ. Эмпирический риск принимает наименьшее возможное значение, равное нулю. Однако этот алгоритм не способен восстановить зависимость вне обучения. Для успешного обучения необходимо не только запоминать, но и обобщать.

Обобщающая способность (generalization ability) метода μ характеризуется величиной $Q(\mu(X^\ell), X^k)$ при условии, что выборки X^ℓ и X^k являются представительными. Для формализации понятия «представительная выборка», как правило, принимается стандартное предположение, что выборки X^ℓ и X^k — простые, полученные из неизвестного вероятностного распределения на множестве X.

Опр. 1.5. *Метод обучения μ называется состоятельным, если при заданных достаточно малых значениях точности ε и надёжности η справедливо неравенство*

$$\mathsf{P}_{X^\ell, X^k}\big\{Q(\mu(X^\ell), X^k) > \varepsilon\big\} < \eta. \tag{1.3}$$

Допустима также эквивалентная формулировка: для любых простых выборок X^ℓ и X^k оценка $Q(\mu(X^\ell), X^k) \leqslant \varepsilon$ справедлива с вероятностью не менее $1 - \eta$.

Обобщающую способность метода μ можно также характеризовать математическим ожиданием ошибки, или *функционалом среднего риска*:

$$R(\mu) = \mathsf{E}_{X^\ell, x}\mathscr{L}(\mu(X^\ell), x) = \mathsf{E}_{X^\ell, X^k}Q(\mu(X^\ell), X^k),$$

однако этот функционал не учитывает разброс (дисперсию) случайной величины Q, следовательно, не даёт гарантированной оценки точности, в отличие от (1.3).

Оценки обобщающей способности позволяют предсказывать качество алгоритмов и строить более надёжные методы обучения. Получение оценок вида (1.3) является фундаментальной и наиболее трудной проблемой статистической теории обучения. Первые оценки были получены в конце 60-х годов советскими математиками Вапником и Червоненкисом [2, 3, 4]. В настоящее время статистическая теория развивается очень активно [14], однако для многих практически интересных случаев оценки обобщающей способности либо неизвестны, либо сильно завышены. Численно точные оценки получены лишь для некоторых частных случаев [20, 18, 23].

Эмпирические оценки обобщающей способности применяются в тех случаях, когда не удаётся воспользоваться теоретическими.

Пусть дана выборка $X^L = (x_i, y_i)_{i=1}^L$. Разобьём её N различными способами на две непересекающиеся подвыборки — обучающую X_n^ℓ длины ℓ и контрольную X_n^k длины $k = L - \ell$. Для каждого разбиения $n = 1, \ldots, N$ построим алгоритм $a_n = \mu(X_n^\ell)$ и вычислим значение $Q_n = Q(a_n, X_n^k)$. Среднее арифметическое значений Q_n по всем разбиениям называется оценкой *скользящего контроля* (cross-validation, CV):

$$\mathrm{CV}(\mu, X^L) = \frac{1}{N}\sum_{n=1}^{N} Q(\mu(X_n^\ell), X_n^k).$$

Возможны различные варианты скользящего контроля, отличающиеся способами разбиения выборки X^L [19], см. также Главу **??**. Чаще всего разбиения генерируются случайным образом, число N берётся в диапазоне от 10 до 100.

В силу независимости выборки матожидание CV совпадает с матожиданием потерь (а в случае бинарной функции потерь — с вероятностью ошибки). Иными словами, оценка скользящего контроля является несмещённой:

$$\mathsf{E}_{X^L}\mathrm{CV}(\mu, X^L) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{\ell}\sum_{i=1}^{\ell}\mathsf{E}_{X_n^\ell, x_i}\mathscr{L}(\mu(X_n^\ell), x_i) = \mathsf{E}_{X^\ell, x}\mathscr{L}(\mu(X^\ell), x).$$

Наряду с эмпирическим средним потерь можно определить эмпирическое распределение потерь, которое при достаточно большом N стремится к выражению в левой части (1.3):

$$\hat{F}_Q(\varepsilon) = \frac{1}{N}\sum_{n=1}^{N}\big[Q(\mu(X_n^\ell), X_n^k) > \varepsilon\big].$$

Это кусочно-постоянная функция от параметра точности ε. В отличие от $\mathrm{CV}(\mu, X^L)$, она даёт представление не только о средней величине потерь, но и о её разбросе. В частности, минимальное и максимальное значения $\big[\min\limits_{n=1..N} Q_n,\ \max\limits_{n=1..N} Q_n\big]$ дают доверительный интервал, в который случайная величина Q_n попадает с вероятностью $\frac{2}{N+1}$. Отсюда следует, что для получения двусторонней оценки с надёжностью 95% достаточно взять $N = 39$ разбиений.

Скользящий контроль является стандартной методикой измерения качества и сравнительного анализа методов обучения. К сожалению, он обладает рядом недостатков. Во-первых, задачу обучения приходится решать N раз, что сопряжено со значительными вычислительными затратами. Во-вторых, оценка скользящего контроля предполагает, что метод μ уже задан. Она ничего не говорит о том, какими свойствами должны обладать «хорошие» методы, и как их строить. Такого рода подсказки дают только теоретические оценки.

Может возникнуть вопрос: коль скоро функционал CV характеризует обобщающую способность, почему бы не строить алгоритм a путём минимизации CV? Увы, бесплатный сыр бывает только в мышеловке. Переобучение появляется вследствие оптимизации параметров модели Γ по конечной выборке. Когда контрольная выборка вовлекается в процесс оптимизации, переобучение возникает снова. На практике скользящий контроль применяется либо для выбора наиболее подходящего метода из небольшого конечного числа альтернатив, либо для подбора одного, но критически важного, параметра метода обучения μ, например, размерности пространства поиска Γ или сложности модели алгоритмов.

1.1.6 Задачи с признаковым описанием объектов

Признаком (feature) называется отображение $f\colon X \to D_f$, описывающее результат измерения некоторой характеристики объекта, где D_f — заданное множество.

В зависимости от множества допустимых значений D_f признаки делятся на следующие типы:

- *бинарный* признак: $D_f = \{0, 1\}$;
- *номинальный* признак: D_f — конечное множество;

- *порядковый* признак: D_f — конечное упорядоченное множество;
- *количественный* признак: $D_f = \mathbb{R}$.

В прикладных задачах встречаются и более сложные случаи. Значениями признаков могут быть числовые последовательности, растровые изображения, функции, графы, результаты запросов к базе данных, и т. д.

Если все признаки имеют одинаковый тип, $D_{f_1} = \cdots = D_{f_n}$, то исходные данные называются *однородными*, в противном случае — *разнородными*.

Пусть имеется набор признаков $f_1, \ldots, f_n$. Вектор $\big(f_1(x), \ldots, f_n(x)\big)$ называют *признаковым описанием* объекта $x \in X$. В дальнейшем мы не будем различать объекты из X и их признаковые описания, полагая $X = D_{f_1} \times \ldots \times D_{f_n}$. Совокупность признаковых описаний всех объектов выборки X^ℓ, записанную в виде таблицы размером $\ell \times n$, называют *матрицей объектов–признаков*:

$$F = \|f_j(x_i)\|_{\ell \times n} = \begin{pmatrix} f_1(x_1) & \ldots & f_n(x_1) \\ \ldots & \ldots & \ldots \\ f_1(x_\ell) & \ldots & f_n(x_\ell) \end{pmatrix}. \tag{1.4}$$

Матрица объектов–признаков является стандартным и наиболее распространённым способом представления исходных данных. Однако на практике встречаются задачи, в которых данные устроены сложнее, например, описания объектов могут иметь переменную длину. В таких случаях по имеющимся исходным данным вычисляются преобразованные данные, имеющие стандартный вид (1.4). Этот приём называется *извлечением признаков* (features extraction) из данных.

Таким образом, признаки — это характеристики объектов, которые либо измеряются непосредственно, либо вычисляются по «сырым» исходным данным. Любое отображение из множества X можно рассматривать как признак. В частности, любой алгоритм $a\colon X \to Y$ также можно рассматривать как признак.

§1.2 Примеры прикладных задач

Прикладные задачи классификации, регрессии и прогнозирования встречаются в самых разных областях человеческой деятельности. Исторически одними из первых были задачи медицинской и технической диагностики. В последнее время количество приложений стремительно возрастает в связи с повсеместным распространением информационных технологий.

1.2.1 Задачи классификации

Пример 1.3. В задачах *медицинской диагностики* в роли объектов выступают пациенты. Признаки характеризуют результаты обследований, симптомы заболевания и применявшиеся методы лечения. Примеры бинарных признаков — пол, наличие головной боли, слабости, тошноты, и т. д. Порядковый признак — тяжесть состояния (удовлетворительное, средней тяжести, тяжёлое, крайне тяжёлое). Количественные признаки — возраст, пульс, артериальное давление, содержание гемоглобина в крови, доза препарата, и т. д. Признаковое описание пациента является, по сути дела, формализованной историей болезни. Накопив достаточное количество прецедентов,

можно решать различные задачи: классифицировать вид заболевания (*дифференциальная диагностика*); определять наиболее целесообразный способ лечения; предсказывать длительность и исход заболевания; оценивать риск осложнений; находить синдромы — наиболее характерные для данного заболевания совокупности симптомов. Ценность такого рода систем в том, что они способны мгновенно анализировать и обобщать огромное количество прецедентов — возможность, недоступная человеку.

Пример 1.4. Задачи *технической диагностики* возникают при управлении сложными техническими комплексами. Например, при мониторинге процесса бурения очень важно вовремя обнаружить предаварийную ситуацию, чтобы остановить процесс и произвести ремонт бурового инструмента. Здесь признаками являются показания датчиков технологических параметров бурения: глубины скважины, оборотов ротора, давления и расхода промывочной жидкости, и т. д. Объекты соответствуют моментам времени, в которые производится регистрация показаний датчиков. Сложность задачи в том, что признаковое описание объекта (ситуации) существенно многомерно. Число различных типов неисправностей также достаточно велико. В этих условиях даже опытному оператору бывает трудно вовремя обнаружить и классифицировать причину неисправности. Для автоматизации мониторинга можно использовать обучаемые алгоритмы. Сначала накапливается обучающая выборка — это те ситуации, которые были выявлены и правильно классифицированы экспертами. Затем по этим прецедентам производится обучение алгоритма распознавания.

Пример 1.5. Задача *предсказания свойств конечной продукции* по свойствам исходных компонент и параметрам технологического процесса. Современные химические, металлургические, текстильные, и другие производства настолько сложны, что построить достаточно точную математическую модель зависимости качества продукции от множества влияющих факторов практически невозможно. В этом случае также прибегают к обучению по прецедентам. Здесь объектами являются экспериментальные партии продукции, изготовленные при фиксированных технологических условиях. Признаками являются параметры исходных компонент и технологического процесса. Целевыми (предсказываемыми) признаками являются показатели качества продукции. Это могут быть качественные показатели — тогда речь идёт о задаче классификации, либо количественные — тогда решается регрессионная задача. Найденные зависимости впоследствии могут использоваться для управления технологическим процессом.

Близкими являются задачи *планирования экспериментов* и *активного обучения* (active learning). Выпуск экспериментальной партии продукции — как правило, дорогое удовольствие. Необходимо так спланировать последовательность экспериментов, чтобы найти оптимальную совокупность технологических параметров, затратив как можно меньше средств на эксперименты. В наших терминах это означает, что формирование каждого последующего объекта обучения происходит с учётом результатов, полученных на предыдущих объектах.

Пример 1.6. В задачах *классификации месторождений полезных ископаемых* признаками являются данные геологической разведки. Наличие или отсутствие тех или иных пород на территории района кодируется бинарными признаками. Физико-химические свойства этих пород могут описываться как количественными, так и качественными признаками. Обучающая выборка составляется из прецедентов двух

классов: районов известных месторождений и районов, в которых интересующее ископаемое обнаружено не было. При поиске редких полезных ископаемых количество объектов может оказаться намного меньше, чем количество признаков. В этой ситуации классические статистические методы не работают, и задача решается путём поиска скрытых логических закономерностей в имеющемся массиве данных. В процессе решения выделяются короткие наборы признаков, обладающие наибольшей информативностью — способностью наилучшим образом разделять классы. По аналогии с медицинской задачей, можно сказать, что отыскиваются «синдромы» месторождений. Это важный побочный результат исследования, представляющий значительный интерес для геофизиков и геологов.

Пример 1.7. Автоматическое *распознавание спама*. Необходимо разделить поток входящих текстовых сообщений (писем электронной почты) на два класса — обычные и нежелательные (спам). Обучающими объектами являются те сообщения, которые конкретный пользователь отклассифицировал сам. Хорошего исходного признакового описания, в отличие от предыдущих задач, здесь не существует, и признаки приходится изобретать «вручную». Примером бинарного признака является наличие определённого ключевого слова или фразы в тексте письма, например, «бесплатно» или «виагра». Наличие китайских иероглифов в теме письма — очень надёжный бинарный признак, особенно, если получатель не знает китайского. Примеры номинальных признаков: страна отправителя, язык сообщения (кодовая страница). Количественные признаки: размер письма в байтах, размер вложения в байтах, число адресатов в рассылке, число спамерских писем, ранее полученных от данного отправителя, число повторений символов _ и -, и т. д. Сложность задачи в том, что система не может самостоятельно генерировать хорошие признаки, не имея того багажа общих знаний, благодаря которому человек легко классифицирует письма.

Пример 1.8. Схожей, но более сложной, является задача *рубрикации текстов*. Она возникает при работе с большим количеством текстовых документов, особенно, если они собираются и используются большим коллективом людей, скажем, в редакции журнала или информационном агентстве. В некоторых случаях от среднего времени поиска документов зависит эффективность работы всей компании.

Системы рубрикации проходили в своём развитии несколько этапов. Когда документов не много, можно без труда помнить, где искать нужный материал. В какой-то момент число документов превосходит критическую массу, и их раскладывают по рубрикам. Затем приходит понимание, что каждому сотруднику нужна своя рубрикация, и, возможно, не единственная. В результате создаётся единое хранилище документов, снабжённое множеством персональных рубрикаторов. При этом возникает потребность автоматизировать процесс раскладывания и поиска документов.

Автоматическое помещение нового документа в нужную рубрику является задачей классификации с огромным количеством классов-рубрик, которые могут пересекаться. Обучающий прецедент создаётся пользователем в тот момент, когда он помещает документ в нужную рубрику. Алгоритм классификации должен уловить логику пользователя и в дальнейшем классифицировать тексты по тем же принципам. В частности, он должен найти критерии, по которым пользователь считает документы схожими. От самого пользователя не требуется в явном виде формулировать эти принципы, более того, он может даже не подозревать об их существовании, пользуясь ими интуитивно.

Один из возможных подходов состоит в том, чтобы сравнивать тексты по составу ключевых слов. Отдельные слова или устойчивые словосочетания считаются ключевыми, если они часто встречаются в небольшом подмножестве документов, и крайне редко — во всех остальных. Обычно это специальные термины, собственные имена, географические названия, и т. д. Документы считаются схожими, если множества их ключевых слов существенно пересекаются. Подмножество ключевых слов считается устойчивым, если оно встречается в значительном числе документов. Эти документы являются схожими и образуют отдельную тематическую рубрику.

Принцип сходства может использоваться и при поиске документов. Типична ситуация, когда пользователь не вполне точно знает, что именно он хочет найти. Часто запрос звучит так: «что ещё известно на эту тему?» или «найти все документы, похожие на данный». Это тоже задача классификации с двумя классами: «подходящие документы» и «остальные документы», и она также может решаться путём оценивания сходства текстов.

Пример 1.9. Задача *распознавания отдельных рукописных символов* решается при автоматической обработке анкет, заполняемых от руки по специальной форме: бланков переписи населения, налоговых деклараций, и.т.д. Это задача классификации с числом классов, равным количеству допустимых символов: букв, цифр и знаков препинания. Изначально сканированное растровое изображение символа — это матрица размером, скажем, 50×50, состоящая из чёрных и белых пикселов. «Сырое» признаковое описание из 2500 бинарных признаков не слишком удобно для классификации. Один из подходов к *извлечению признаков* заключается в том, чтобы разбить изображение на некоторое количество областей (возможно, пересекающихся) и подсчитать долю чёрных пикселов в каждой области. Такие признаки более информативны и уже позволяют делать некоторые выводы. Например, если чёрные пиксели группируются по горизонтали вверху и по вертикали в центре, то символ похож на букву «Т». Другой подход заключается в том, чтобы по растру построить контур изображения в виде ломаной линии, и в качестве признаков взять характеристики этой ломаной, например, положение, наклон и длину каждого звена. Построение таких алгоритмов требует проведения большой экспериментальной работы по отбору признаков.

Пример 1.10. Задачи *идентификации подписей* бывают двух типов: off-line и on-line. В случае off-line подпись вводится в компьютер с помощью сканера и представляет собой растровое изображение, поэтому задача во многом схожа с предыдущей. В случае on-line человек вводит подпись с помощью электронного пера. Эта задача считается более простой по сравнению с off-line, поскольку, кроме самого изображения подписи, в распоряжении распознающей системы оказывается существенная дополнительная информация о последовательности и скорости движения пера. Таким образом, объекты (подписи) описываются не растровым изображением, а парой функций времени $x(t)$ и $y(t)$, представляющих координаты пера. В данной задаче целесообразно вообще не переходить к признакам, а сравнивать подписи непосредственно, как временные ряды. Задача сводится к построению такой меры сходства, при которой все подписи, принадлежащие одному человеку, оказывались бы схожими, а подписи разных людей существенно различались.

Пример 1.11. Задача *распознавания слитной речи* является, пожалуй, одной из самых трудных среди задач классификации и распознавания образов. Несмотря на значительные усилия, прилагаемые во многих лабораториях по всему миру, пока не удаётся построить алгоритм, способный распознавать человеческую речь с приемлемым для коммерческих приложений уровнем ошибок. Сложность задачи в том, что отдельные звуковые сигналы могут существенно видоизменяться в зависимости от положения звука в слове, слова в предложении, внешних шумов и особенностей речи диктора. Успешное распознавание речи практически невозможно без использования знаний из нескольких смежных областей: физики (акустики), лингвистики (фонетики), теории классификации, теории марковских цепей.

В отличие от «сверхзадачи» преобразования слитной речи в текст, *распознавание голосовых команд* из заданного списка или *распознавание ключевых слов* в слитной речи являются относительно простыми задачами. Вообще, задачи классификации существенно упрощаются при уменьшении количества классов.

Пример 1.12. Задача *оценивания заёмщиков* решается банками при выдаче кредитов. Потребность в автоматизации процедуры выдачи кредитов впервые возникла в период бума кредитных карт 60-70-х годов в США и других развитых странах. Объектами в данном случае являются заёмщики — физические или юридические лица, претендующие на получение кредита. В случае физических лиц признаковое описание состоит из анкеты, которую заполняет сам заёмщик, и, возможно, дополнительной информации, которую банк собирает о нём из собственных источников. Примеры бинарных признаков: пол, наличие телефона. Номинальные признаки — место проживания, профессия, работодатель. Порядковые признаки — образование, занимаемая должность. Количественные признаки — возраст, стаж работы, доход семьи, размер задолженностей в других банках, сумма кредита. Обучающая выборка составляется из заёмщиков с известной кредитной историей. В простейшем случае принятие решений сводится к классификации заёмщиков на два класса: «хороших» и «плохих». Кредиты выдаются только заёмщикам первого класса. В более сложном случае оценивается суммарное число баллов (score) заёмщика, набранных по совокупности информативных признаков. Чем выше оценка, тем более надёжным считается заёмщик. Отсюда и название — *кредитный скоринг* (credit scoring). На стадии обучения производится синтез и отбор информативных признаков и определяется, сколько баллов назначать за каждый признак, чтобы риск принимаемых решений был минимален. Следующая задача — решить, на каких условиях выдавать кредит: определить процентную ставку, срок погашения, и прочие параметры кредитного договора. Эта задача также сводится к обучению по прецедентам.

Пример 1.13. Задача *оценивания привлекательности инвестиционных проектов* сходна с задачей оценивания заёмщиков. С ней сталкиваются инвестиционные компании, банки, венчурные фонды, крупные корпорации, финансирующие исследовательские, инновационные и рискованные бизнес-проекты. Объектами являются заявки на проекты, представляемые в виде анкет. Типичное количество пунктов анкеты — около сотни, количество анкет — сотни или тысячи. Основная сложность данной задачи — низкая степень формализованности анкет. Наиболее важная с точки зрения экспертов информация заключается, как правило, в нескольких текстовых полях, содержательно описывающих суть проекта. Разумеется, полная автоматизация

процесса рассмотрения заявок невозможна. Однако можно решать задачу предварительной классификации заявок на «хорошие» — достойные внимательного изучения экспертами, и «плохие» — которые с высокой вероятностью не пройдут экспертный отбор. Плохие заявки можно отсеять заранее и полностью автоматически, значительно сократив загруженность экспертов. Обучающая выборка составляется из ранее классифицированных анкет, например, заявок прошлых лет. Методы решения задачи аналогичны методам оценивания заёмщиков.

Как и в других задачах анализа текстов (обнаружения спама, рубрикации), наиболее сложным является этап предварительной обработки данных, в ходе которого создаются числовые признаки заявки. Признаками могут быть как исходные характеристики заявки: срок выполнения проекта, объем требуемых инвестиций, доходность, рентабельность, количество исполнителей, отрасль, и т. д., так и вычисленные характеристики, отражающие качество заполнения анкеты: объем аннотации и других текстовых полей, степень заполнения полей, соответствие анкеты формальным требованиям, и т. д.

Пример 1.14. Задача *предсказания ухода клиентов* (churn prediction) возникает у крупных и средних компаний, работающих с большим количеством клиентов, как правило, с физическими лицами. Особенно актуальна эта задача для современных телекоммуникационных компаний. Когда рынок приходит в состояние, близкое к насыщению, основные усилия компаний направляются не на привлечение новых клиентов, а на удержание старых. Для этого необходимо как можно точнее выделить класс клиентов, склонных к уходу в ближайшее время. Классификация производится на основе информации, хранящейся у компании: клиентских анкет, данных о частоте пользования услугами компании, составе услуг, тарифных планах, регулярности платежей, и т. д. Наиболее информативны данные о том, что именно изменилось в поведении клиента за последнее время. Поэтому объектами, строго говоря, являются не сами клиенты, а пары «клиент x_i в момент времени t_i». Требуется предсказать, уйдёт ли клиент к моменту времени $t_i + \Delta t$. Обучающая выборка формируется из клиентов, о которых доподлинно известно, в какой момент они ушли.

Данная задача характеризуется высоким уровнем ошибок, так как предсказывать поведение людей очень непросто. Задача ставится более прагматично — минимизировать суммарные потери компании. Для этого выделяется относительно небольшая целевая группа клиентов с наибольшей вероятностью ухода и наибольшей вероятностью того, что клиент изменит своё решение после предоставления ему скидки или дополнительных услуг. Учитывается также соотношение стоимости маркетинговых воздействий и перспективного дохода от клиента.

Пример 1.15. Задачи *обнаружения мошенничества* (fraud detection).

Пример 1.16. Задачи *составления психологических тестов*.

1.2.2 Задачи восстановления регрессии

Пример 1.17. Термин «регрессия» был введён в 1886 году антропологом Фрэнсисом Гальтоном при изучении статистических закономерностей наследственности роста. Повседневный опыт подсказывает, что в среднем рост взрослых детей тем

больше, чем выше их родители. Однако Гальтон обнаружил, что сыновья очень высоких отцов часто имеют не столь высокий рост. Он собрал выборку данных по 928 парам отец-сын. Количественно зависимость неплохо описывалась линейной функцией $y = \frac{2}{3}x$, где x — отклонение роста отца от среднего, y — отклонение роста сына от среднего. Гальтон назвал это явление «регрессией к посредственности», то есть к среднему значению в популяции. Термин *регрессия* — движение назад — намекал также на нестандартный для того времени ход исследования: сначала были собраны данные, затем по ним угадана модель зависимости, тогда как традиционно поступали наоборот: данные использовались лишь для проверки теоретических моделей. Это был один из первых случаев моделирования, основанного исключительно на данных. Позже термин, возникший в частной прикладной задаче, закрепился за широким классом методов восстановления зависимостей.

Огромное количество регрессионных задач возникает в физике и технике.

1.2.3 Задачи прогнозирования и принятия решений

Пример 1.18. Задача *прогнозирования потребительского спроса* решается современными супермаркетами и торговыми розничными сетями. Для эффективного управления торговой сетью необходимо прогнозировать объёмы продаж для каждого товара на заданное число дней вперёд. На основе этих прогнозов осуществляется планирование закупок, управление ассортиментом, формирование ценовой политики, планирование промоакций (рекламных кампаний). Специфика задачи в том, что количество товаров может исчисляться десятками или даже сотнями тысяч. Прогнозирование и принятие решений по каждому товару «вручную» просто немыслимо. Исходными данными для прогнозирования являются временны́е ряды цен и объёмов продаж по товарам и по отдельным магазинам. Современные технологии позволяют снимать эти данные непосредственно с кассовых аппаратов. Для увеличения точности прогнозов необходимо также учитывать различные внешние факторы, влияющие на потребительский спрос: уровень инфляции, погодные условия, рекламные кампании, социально-демографические условия, активность конкурентов. В зависимости от целей анализа в роли объектов выступают либо товары, либо магазины, либо пары «магазин–товар». Ещё одна особенность задачи — несимметричность функции потерь. Если прогноз делается с целью планирования закупок, то потери от заниженного прогноза, как правило, существенно выше потерь от завышенного.

Пример 1.19. *Принятие инвестиционных решений* на финансовом рынке — это та задача, в которой умение хорошо прогнозировать самым непосредственным образом превращается в прибыль. Если инвестор предполагает, что цена акции вырастет, он покупает акции, надеясь продать их позже по более высокой цене. И, наоборот, прогнозируя падение цен, инвестор продаёт акции, чтобы впоследствии выкупить их обратно по более низкой цене. Задача инвестора-спекулянта в том, чтобы правильно предугадать направление будущего изменения цены — роста или падения. При этом инвестор сильно рискует: известно, что на финансовом рынке с положительной прибылью остаются только около 30% игроков. Тем не менее, инвесторы готовы брать на себя этот риск в надежде на высокую прибыль. Принято считать, что все вместе они выполняют важную экономическую функцию — обеспечивают

ликвидность рынка, то есть возможность в любой момент купить или продать достаточно крупный пакет акций.

Большой популярностью пользуются автоматические торговые стратегии — алгоритмы, принимающие торговые решения без участия человека. Разработка такого алгоритма — это задача обучения по прецедентам. В роли объектов выступают ситуации, фактически, моменты времени. Описание объекта — это предыстория изменения цен и объёмов торгов, зафиксированная к данному моменту. В простейшем случае объекты необходимо классифицировать на три класса, соответствующих возможным решениям: купить, продать или выжидать. Обучающей выборкой для настройки торговых стратегий служат исторические данные о движении цен и объёмов за некоторый промежуток времени. Критерий качества в данном случае существенно отличается от стандартного функционала средней ошибки прогнозов или классификаций, поскольку инвестора интересует не точность прогнозирования, а максимизация итоговой прибыли. Современный биржевой *технический анализ* насчитывает сотни параметрических торговых стратегий, которые принято настраивать по критерию максимума прибыли на выбранном интервале истории.

Биржевая торговля имеет и обратную сторону. Задачи обучения по прецедентам возникают также и у самой биржи, как организатора торгов, главным образом, при проведении *биржевого надзора* (market surveillance). Цель надзора — поддержание справедливого и эффективного рынка путём выявления и расследования случаев умышленного манипулирования ценами и инсайдерской торговли. Налаженная работа службы надзора, периодически сопровождающаяся «показательными судами», значительно повышает доверие к бирже у основной массы рядовых инвесторов. В мировой практике известны случаи, когда введение надзора позволяло торговым площадкам резко увеличить объёмы торгов.

Пример 1.20. Задача *выявления интервенций на финансовом рынке* и оценивания их влияния на цену. Практически ни одна биржевая манипуляция не обходится без финансовых интервенций. *Интервенция* — это когда некоторый участник торгов (или несколько участников по сговору) проводит кратковременные однонаправленные операции на рынке: либо скупает крупный пакет, что приводит к росту цены, либо продаёт крупный пакет, что приводит к падению цены. Интервенцию можно проводить по-разному. «Мягкая» интервенция, собственно таковой не является, так как её цель — действительно продать или купить крупный пакет, например, в интересах крупного инвестора, переводящего свои капиталы в другую отрасль. Мягкая интервенция проводится так, чтобы по возможности не оказать давления на рынок. При этом трейдеры «аккуратно» распределяют свои операции во времени, действуя многократно мелкими партиями, после каждой операции позволяя рынку вернуться в равновесное состояние. Это не является противозаконным или подозрительным. «Жёсткая» интервенция, наоборот, преследует цель максимально сдвинуть цену в направлении, выгодном манипулятору, затратив на это минимальный объём средств. Как правило, она проводится довольно быстро, в тщательно подобранный момент времени, когда рынок наиболее слаб, и является частью заранее спланированного сценария. Если манипуляторам удаётся реализовать свой сценарий, рядовые инвесторы терпят значительные убытки. Манипуляции не выгодны рынку в целом и должны преследоваться по закону.

Для автоматического обнаружения интервенций в потоке биржевых данных необходимо иметь модель интервенции. Параметрическая модель описывает зависимость результирующего скачка цены на интервале времени от действий подозреваемого участника. Определение параметров такой модели — это типичная задача обучения по прецедентам. Здесь объектами являются достаточно короткие интервалы времени, в течение которых один из участников торгов совершил значительный объём однонаправленных операций. Признаковое описание объекта состоит примерно из десятка величин, характеризующих действия подозреваемого участника на данном интервале. В частности: сальдо операций участника (т. е. разность объёма его покупок и продаж), доля участника в торговом обороте, распределённость операций участника во времени, полученная участником прибыль, и другие. Целевой величиной является скачок цены, вычисляемый как разность средних цен после предполагаемой интервенции и до неё.

Если зависимость скачка цены от признакового описания интервенции будет найдена, то наличие интервенции на предъявленном интервале времени можно будет обосновывать статистически, например, так: «в 95% аналогичных случаев скачок цены значимо отличался от нуля и составлял $+(17 \pm 10)$ пунктов». Регрессия позволяет количественно оценить, насколько сильно действия подозреваемого участника повлияли на цену, и указать доверительный интервал для этого влияния.

Пример 1.21. Задача *выявления нестандартного поведения* участников торгов во многом схожа с выявлением интервенций. Отличие в том, что критерии нестандартности в данном случае формулируются экспертами заранее, в виде простых понятных правил. Например: «сделка перекрёстная (с самим собой), и её объём превышает $\varepsilon_1\%$ от среднего дневного оборота, и её цена отклоняется от средней на $\varepsilon_2\%$ или более». Задача подбора порогов $\varepsilon_1, \varepsilon_2$ — это типичная задача классификации. Обучающая выборка составляется из проверенных случаев стандартного (класс 1) и нестандартного (класс 0) поведения. Кроме того, накладываются дополнительные ограничения на среднее количество сигналов, выдаваемых в течение торгового дня. Их не должно быть слишком много, чтобы эксперты службы надзора имели физическую возможность расследовать найденные случаи вручную. Считается, что в данной задаче полная автоматизация невозможна, так как расследование каждого случая требует привлечения неформализованных знаний экспертов.

1.2.4 Задачи кластеризации

Задачи кластеризации или обучения без учителя отличаются от задач классификации тем, что в них не задаются ответы $y_i = y^*(x_i)$. Известны только сами объекты x_i, и требуется разбить выборку на непересекающиеся подмножества (кластеры) так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались. Число кластеров может быть известно заранее, но чаще требуется определить и его.

Пример 1.22. Основным инструментом *социологических и маркетинговых исследований* является проведение опросов. Чтобы результаты опроса были объективны, необходимо обеспечить представительность выборки респондентов. С другой стороны, требуется минимизировать стоимость проведения опроса. Поэтому при *планировании опросов* возникает следующая задача: отобрать как можно меньше респон-

дентов, чтобы они образовывали *репрезентативную выборку*, то есть представляли весь спектр общественного мнения. Например, при формировании множества точек опроса (это могут быть города, районы, магазины, и т. д.) сначала составляются признаковые описания достаточно большого числа точек. Это можно сделать, используя недорогие способы сбора информации — пробные опросы и/или фиксацию характеристик самих точек. Затем решается задача кластеризации, и из каждого кластера отбирается по одной представительной точке. Только в отобранном множестве точек производится основной, наиболее ресурсоёмкий, опрос.

Пример 1.23. Задача *тематической кластеризации новостей*. В настоящее время новостные потоки в сети Интернет стали настолько огромными, что ориентироваться и оценивать поступающую информацию становится очень тяжело, а часто практически невозможно. Одна новость может породить цепочку новостей, в значительной степени дублирующих друг друга. Поэтому наряду с нахождением всех похожих новостей ставятся задачи определения самого первого сообщения, определения связей между сообщениями и, в конечном итоге, отслеживания всей цепочки событий. Выделенный кластер сообщений заменяется не одним типичным представителем, а довольно сложной древовидной структурой, описывающей сценарий развития событий. Это описание должно быть одновременно полным и неизбыточным.

1.2.5 Задачи анализа клиентских сред

Клиентская среда — это совокупность клиентов (субъектов, user), регулярно пользующихся фиксированным набором сервисов или ресурсов (объектов, item). Клиентские среды возникают в самых разных сферах бизнеса, и не только бизнеса. Можно говорить о клиентских средах производителей товаров, дилерских сетей, сетей супермаркетов, операторов связи, эмитентов пластиковых карт, библиотек, электронных магазинов, интернет-порталов, и т. д.

Современные информационные технологии позволяют детально протоколировать действия клиентов в электронном виде. Эти протоколы содержат ценную информацию, необходимую для решения широкого спектра задач, объединяемых модным термином *управление взаимоотношениями с клиентами* (Customer Relationship Management, CRM). Оно подразумевает выявление целевых групп клиентов, персонализацию работы с клиентами, в частности, формирование персональных предложений, прогнозирование возможного оттока клиентов (churn prediction), выявление мошенничества (fraud detection) или необычного поведения клиентов. Решение этих и других аналитических задач направлено в конечном итоге на повышение качества сервисов, автоматизацию обратной связи с клиентом, более эффективное привлечение и удержание клиентов.

Пример 1.24. Задача *предсказания рейтингов* решается в интернет-магазинах, особенно книжных, и сетях видеопроката. Приобретая через сайт некоторый товар (книгу, фильм, и т. д.), клиент имеет возможность выразить своё отношение к нему, выставив рейтинг, обычно целое число от 1 до 5. Система использует информацию о всех выставленных рейтингах для *персонализации* предложения. Когда клиент видит на сайте страницу с описанием товара, ему показывается также список всех схожих товаров, получивших высокий рейтинг у схожих клиентов. Основная задача — быстро находить в огромном объёме данных множества схожих клиентов и схо-

жих товаров и прогнозировать их рейтинги для данного клиента. Формально задачу можно ставить и как задачу классификации. Роль матрицы объектов–признаков играет матрица клиентов–товаров, заполненная значениями рейтингов. Как правило, она сильно разрежена и имеет более 90% пустых ячеек. Фиксированного целевого признака в этой задаче нет. Алгоритм классификации должен уметь предсказать рейтинг для любой незаполненной ячейки матрицы.

О сложности и актуальности этой задачи говорит следующий факт. В октябре 2006 года крупнейшая американская компания Netflix, занимающаяся видеопрокатом через Internet, объявила международный конкурс[2] с призом в 1 миллион долларов тому, кто сможет на 10% улучшить точность прогнозирования рейтингов, по сравнению с системой Cinematch, эксплуатируемой в Netflix. Примечательно, что прогнозы самой Cinematch были лишь на те же 10% точнее элементарных средних рейтингов фильмов. Компания крайне заинтересована в увеличении точности прогнозов, поскольку около 70% заказов поступают через рекомендующую систему.

Пример 1.25. Задача *персонализации поиска*. Интернет в его современном состоянии нередко называют «информационной помойкой». Информации настолько много, что поисковые машины требуют чрезмерно чётко формулировать критерий поиска. К сожалению, сузить запрос можно десятком возможных способов, некоторые из которых пользователь может и не знать. Хотелось бы, чтобы система сама узнавала пользователя по его прошлому поведению, и ранжировала результаты поиска персонально для него, ориентируясь как на круг его обычных интересов, так и на его последние запросы и посещения, отражающие его актуальный интерес. Например, разные пользователи могут сделать запрос "`churn prediction`", однако специалиста IT-департамента будут интересовать вопросы внедрения существующих на рынке систем управления оттоком клиентов, менеджера по продажам — презентации компаний-конкурентов, учёного — алгоритмы классификации с применением в бизнес-аналитике, а студента — обзоры и рефераты на эту тему. Возможное решение проблемы основано на идее персонализации. Результаты поиска сортируются так, чтобы в начале списка оказались документы, схожие с теми, которые смотрели схожие пользователи. Самое сложное в этой задаче — построить адекватные меры сходства. Пользователи схожи, если они смотрят схожие документы. Документы схожи, если их смотрят схожие пользователи. Принцип взаимного согласования двух мер сходства лежит в основе технологии *анализа клиентских сред* (АКС), разработанной в компании Форексис[3]. Насколько адекватны построенные меры сходства, можно судить, например, по *картам сходства* сайтов (Рис. 1). Сайты, близкие на карте, как правило, оказываются схожими по тематике, хотя для оценивания сходства содержимое документов вообще не использовалось, учитывались только данные о посещениях.

Пример 1.26. Довольно неожиданная область применения АКС — *анализ результатов голосования* на парламентских выборах. Здесь в роли объектов выступают политические партии; субъектами являются регионы или избирательные участки; рейтинг объекта есть число голосов, отданных на данном участке за данную пар-

[2] http://www.netflixprize.com.

[3] http://www.forecsys.ru.

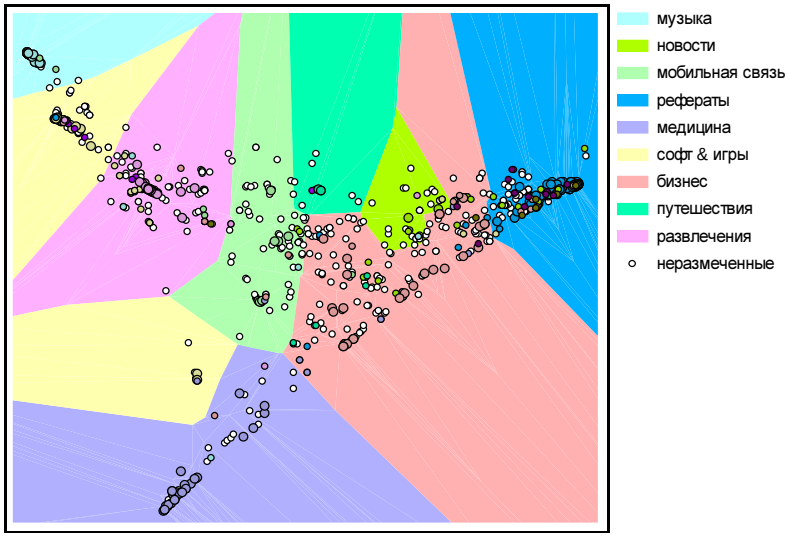

Рис. 1. Карта сходства ресурсов Интернет, построенная методом многомерного шкалирования, см. **??**. Исходные данные, предоставленные компанией Яндекс, содержали протоколы посещения 129 600 ресурсов 14 606 пользователями. На карте показаны только 1000 наиболее посещаемых ресурсов, из них 400 были разделены по тематике на 9 классов. Информация о тематике не использовалась для вычисления сходства и размещения точек на карте, а только для раскраски карты.

тию. Несмотря на существенные содержательные отличия, математические формулировки возникающих здесь задач те же, что и в случаях «документов в Интернете» или «товаров в магазине». Определяются меры сходства между партиями и между регионами. Регионы схожи, если они имеют похожие распределения голосов по партиям. Партии схожи, если они имеют похожие распределения голосов по регионам. Полученные меры сходства позволяют отвечать, например, на следующие вопросы: какие партии близки и делят электорат между собой; какие группы партий образуют «политические полюса»; сколько этих полюсов; какие регионы к какому полюсу тяготеют; в каких регионах есть шансы перетянуть голоса в сторону той или иной партии.

В анализе клиентских сред функции сходства (метрики), построенные одновременно и для субъектов (клиентов), и для объектов (ресурсов, товаров, сервисов), позволяют решать целый спектр прикладных задач. Метрика на клиентах используется для классификации и кластеризации клиентов, поиска *единомышленников* (like-minded people), обнаружения необычного поведения клиентов. Метрика на ресурсах позволяет лучше структурировать ассортимент путём кластеризации, находить сопутствующие и взаимозаменяемые товары. При формировании персонального предложения приходится использовать обе метрики.

§1.3 Эвристические принципы обучения по прецедентам

Несмотря на большое разнообразие применяемых моделей алгоритмов и методов их настройки, общих принципов их построения не так уж много. Наиболее удачные модели совмещают в себе сразу несколько принципов.

Все эти принципы являются в той или иной степени *эвристическими* — они опираются не только на строгие математические обоснования, но в значительной степени на соображения здравого смысла. Не существует универсальных моделей,

подходящих под любые задачи. Каждая эвристика хорошо работает лишь в своём классе задач. Поэтому мы изучаем разнообразные эвристические принципы и приёмы. Понимание взаимосвязей между ними позволяет сочетать различные эвристики и конструировать новые методы, наиболее подходящие для конкретных случаев.

Ниже мы дадим краткий обзор основных эвристик. Для некоторых из них будет показан простейший вариант метода обучения. Простейшие методы обладают массой недостатков, обсуждению и устранению которых будет посвящён весь остальной курс. В то же время, они наиболее ярко иллюстрируют основные идеи.

1.3.1 Принцип сходства

Принцип сходства предполагает, что на множестве X можно так ввести функцию расстояния между объектами, что близким объектам будут, как правило, соответствовать близкие ответы. Применительно к задачам восстановления регрессии это равносильно предположению, что целевая функция y^* является достаточно гладкой. Даже если она имеет резкие скачки, они не могут находиться повсюду, как у функции Дирихле. В случае классификации принцип сходства означает, что схожие объекты, как правило, лежат в одном классе. Граница классов может быть довольно сильно изрезана, но она не может проходить везде, как кривая Пеано. В «хорошей» задаче классы представляют собой области, компактно расположенные в пространстве X. Это предположение называют *гипотезой компактности*. Эмпирический опыт убеждает, что столь сложные математические модели, как функция Дирихле или кривая Пеано, просто не встречаются в природе — «Бог изощрён, но не злонамерен».

Принцип сходства лежит в основе *метрических алгоритмов* классификации. Пусть $\rho(x, x')$ — метрика в пространстве объектов. Метод *ближайшего соседа* (nearest neighbor) относит объект x к тому классу, которому принадлежит ближайший объект обучающей выборки X^ℓ:

$$a(x) = y_{n(x)}, \quad \text{где} \quad n(x) = \arg\min_{i=1,\dots,\ell} \rho(x, x_i).$$

Это, пожалуй, самой простой из всех алгоритмов классификации. В нём нет настраиваемых параметров. Обучение сводится к элементарному запоминанию выборки. Стремление обогатить эту модель параметрами приводит к широкому классу метрических алгоритмов классификации, которые рассматриваются в **??**. На принципе близости основаны методы кластеризации, непараметрической регрессии, многомерного шкалирования. Все они подробно обсуждаются в главе **??**.

Наиболее тонкий вопрос для всех метрических алгоритмов анализа данных — как построить метрику ρ. Если объекты представлены своими признаковыми описаниями, то можно взять евклидово расстояние между объектами:

$$\rho^2(x, x') = \sum_{j=1}^{n} \big(f_j(x) - f_j(x')\big)^2,$$

однако это далеко не единственный вариант, и далеко не самый лучший. Проблема выбора метрики обсуждается в разделе **??**.

1.3.2 Принцип минимизации эмпирического риска

Эмпирическим риском принято называть среднюю ошибку алгоритма на обучающей выборке $Q(a, X^\ell)$. Принцип *минимизации эмпирического риска* заключается в том, чтобы в заданной модели A найти алгоритм, допускающий наименьшее число ошибок на обучающей выборке X^ℓ:

$$\mu(X^\ell) = \arg\min_{a \in A} Q(a, X^\ell).$$

Как уже упоминалось в 1.1.3, данная задача решается с помощью численных методов оптимизации. В редких случаях её удаётся решить аналитически.

Наиболее сильным вариантом минимизации эмпирического риска является *принцип корректности*, который требует, чтобы искомый алгоритм вообще не допускал ошибок на обучающей выборке:

$$Q(\mu(X^\ell), X^\ell) = 0.$$

Требование корректности лежит в основе алгебраического подхода к проблеме распознавания, см. **??**.

Пример 1.27. Частным случаем восстановления регрессии является классическая задача о приближении функций. Требуется построить функцию $a(x)$, проходящую через заданные точки $(x_1, y_1), \ldots, (x_\ell, y_\ell)$. Если условия $a(x_i) = y_i$ должны удовлетворяться точно (требование корректности), то это задача *интерполяции*. Если же достаточно приближённого соответствия, то это задача *аппроксимации*.

Применение полиномиальной модели алгоритмов $a(x, \gamma) = \gamma_0 + \gamma_1 x + \cdots + \gamma_n x^n$, квадратичной функции потерь $L(y, y') = (y - y')^2$ и принципа минимизации эмпирического риска приводит к классической задаче аппроксимации, которая решается методом наименьших квадратов, см. **??**:

$$Q(\gamma, X^\ell) = \sum_{i=1}^{\ell} (\gamma_0 + \gamma_1 x_i + \cdots + \gamma_n x_i^n - y_i)^2 \to \min_{\gamma_0, \ldots, \gamma_n}.$$

Полиномиальная модель склонна к *переобучению* при увеличении степени полинома. Рассмотрим для примера гладкую функцию $y^*(x) = (1 + 25x^2)^{-1}$ на отрезке $[-2, 2]$. По мере увеличения n средняя ошибка на обучающей выборке

$$X^\ell = \left\{ x_i = 4\tfrac{i-1}{\ell-1} - 2 \;\middle|\; i = 1, \ldots, \ell \right\}$$

монотонно уменьшается. При этом средняя ошибка на контрольных данных

$$X^k = \left\{ x_i = 4\tfrac{i-0.5}{\ell-1} - 2 \;\middle|\; i = 1, \ldots, \ell - 1 \right\}$$

сначала уменьшается, затем резко возрастает, Рис. 2. Переобучение проявляется в виде резкого ухудшения качества аппроксимации на концах отрезка, см. Рис. 3.

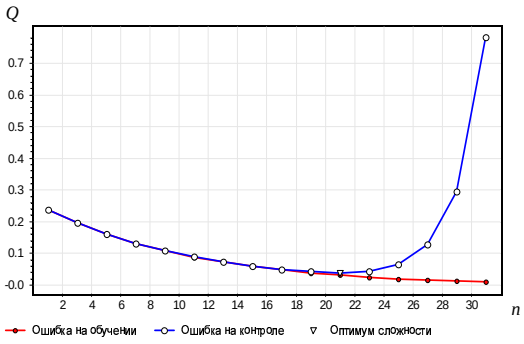

Рис. 2. Зависимость средней ошибки на обучении и контроле от степени полинома, $\ell = 50$.

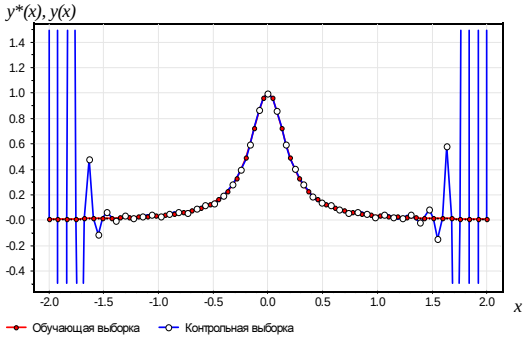

Рис. 3. Проявление эффекта переобучения при аппроксимации полиномом, $n = 40$, $\ell = 50$.

1.3.3 Принцип регуляризации

Переобучение часто возникает при использовании чрезмерно сложных моделей алгоритмов. Модели, обладающие избыточным числом свободных параметров, позволяют точнее воспроизводить ответы на материале обучения. Однако попытка описать обучающие данные точнее, чем в принципе позволяет суммарная погрешность измерений и самой модели, может привести к катастрофическому снижению обобщающей способности. Именно это и наблюдалось в примере 1.27. На практике любая модель не точна, поэтому проблема переобучения носит всеобщий характер в машинном обучении.

Известный философский принцип *бритвы Оккама* (Occam's razor) гласит, что из множества допустимых решений всегда следует выбирать наиболее простое. В частности, модель алгоритмов не должна иметь избыточных параметров.

Справедливости ради отметим, что применение сложных моделей не всегда ведёт к переобучению. Известны методы, позволяющие находить достаточно надёжные алгоритмы в очень сложных алгоритмических моделях, например, методы обучения алгоритмических композиций, о которых пойдёт речь в главе **??**. Сложность модели в наших задачах — довольно тонкая характеристика. Это количество алгоритмов в модели, но не всех, а только тех, которые могут быть получены в результате обучения. То есть сложность завит не только от модели алгоритмов, но и от восстанавливаемой зависимости, и от метода обучения, и даже от самой обучающей выборки. Увеличение числа параметров модели не влечёт повышение сложности, если в процессе настройки на эти параметры накладываются определённые ограничения.

Один из способов ограничения сложности состоит в том, чтобы отойти от принципа минимизации эмпирического риска и добавить к функционалу $Q(a, X^\ell)$ штрафное слагаемое, наказывающее чрезмерно сложные модели:

$$\mu(X^\ell) = \arg\min_{a \in A} \big(Q(a, X^\ell) + \tau C(a)\big),$$

где число τ называется *параметром регуляризации*, функционал $C(a)$ выражает сложность алгоритма a. Это и есть принцип регуляризации некорректно поставленных задач по А. Н. Тихонову [11]: если решение существует, но оно не единственно или не устойчиво, то из множества возможных решений следует выбрать такое, которое минимизирует дополнительный критерий регуляризации $C(a)$.

Существует масса способов задать штрафное слагаемое $C(a)$. Простейшая эвристика — взять в качестве $C(a)$ число настраиваемых параметров алгоритма a, как это делается в информационных критериях AIC и BIC, см. **??**. В алгоритмах классификации и регрессии, линейных по вектору параметров γ, часто применяется другая эвристика — взять в качестве штрафа норму этого вектора: $C(a) = \|\gamma\|$. Существуют и другие разновидности штрафных функционалов, основанные на теоретических оценках обобщающей способности.

1.3.4 Принцип разделимости

Принцип разделимости относится к задачам классификации. Он предполагает, что объекты в пространстве X могут быть разделены некоторой поверхностью. Например, *линейная разделимость* двух классов в евклидовом пространстве X означает, что существует гиперплоскость, относительно которой точки одного класса лежат по одну сторону, а точки второго класса — по другую.

Пусть $Y = \{-1, +1\}$, и объекты описываются признаками $f_1, \ldots, f_n$. *Линейным разделяющим правилом* называется алгоритм классификации вида

$$a(x) = \operatorname{sign}\big(\alpha_1 f_1(x) + \cdots + \alpha_n f_n(x)\big),$$

где весовые коэффициенты $\alpha_1, \ldots, \alpha_n$ являются параметрами алгоритма и настраиваются по обучающей выборке X^ℓ.

Самый простой метод настройки весов — *персептронный алгоритм*. Он работает следующим образом. Сначала веса инициализируются небольшими случайными значениями, например, в интервале $\left(-\frac{1}{n}, +\frac{1}{n}\right)$. Затем по очереди просматриваются все объекты обучающей выборки. Если объект $x_i \in X^\ell$ классифицируется неверно, $a(x_i)y_i < 0$, то веса корректируются:

$$\alpha_j := \alpha_j + f_j(x_i)y_i, \quad j = 1, \ldots, n.$$

Американский учёный Новиков в 1962 году доказал, что если выборка изначально линейно разделима, то этот алгоритм сходится за конечное число шагов — рано или поздно обучающая выборка окажется разделена на два класса без ошибок. Именно с этой теоремой принято связывать появление математической теории распознавания образов.

Ещё один принцип построения разделяющих поверхностей связан со *статистическим байесовским подходом* (глава **??**). Оказывается, вид оптимальной разделяющей поверхности является следствием тех или иных предположений о виде функций распределения классов. Например, если предположить, что каждый класс описывается многомерным нормальным распределением, то разделяющая поверхность будет квадратичной. Если к тому же все классы имеют одинаковую форму (равные ковариационные матрицы), то квадратичная поверхность вырождается в линейную, и мы снова приходим к линейному разделяющему правилу. В этом случае метод настройки весов называется *линейным дискриминантом Фишера* — см. **??**.

Принцип разделимости лежит в основе широко известного *метода опорных векторов* (SVM). Он исходит из дополнительного требования, чтобы расстояние от разделяющей поверхности до ближайших объектов выборки было максимальным, см. главу **??**. Известно, что максимизация *зазора* (margin) между классами способствует более уверенной классификации и улучшает обобщающую способность.

Неявно принцип разделимости присутствует всегда, когда алгоритм классификации строится в виде $a(x) = C\big(b(x)\big)$, где функция $b(x)$ даёт числовую оценку принадлежности объекта классу и называется *алгоритмическим оператором* или *вещественнозначным классификатором* (real-valued classifier); функция $C(b)$ переводит оценку принадлежности собственно в номер класса и называется *решающим правилом*. Обычно C имеет очень простой вид. Например, в случае $Y = \{-1, +1\}$ естественно выбрать функцию $C(b) = \text{sign}(b)$.

Если $Y = \{-1, +1\}$, то величина $m_i = b(x_i)y_i$ называется *отступом* (margin) объекта x_i от поверхности, разделяющей классы. Отступ m_i отрицателен тогда и только тогда, когда алгоритм допускает ошибку на объекте x_i. Распределение отступов обучающих объектов характеризуют геометрию взаимного расположения классов. Аккуратный учёт этой важной дополнительной информации способствует повышению качества классификации [16, 17].

Развитие метода опорных векторов привело к принципу *явной максимизации отступов* (direct optimization of margin) [21, 22]. Дискретная функция потерь, стандартная для задач классификации, выражает только факт наличия ошибки:

$$\mathscr{L}(b, x_i) = [b(x_i)y_i < 0] = [m_i < 0].$$

Вместо неё можно вводить гладкие функции потерь, учитывающие не только знак, но и величину отступа, например, экспоненциальную функцию $\mathscr{L}(b, x_i) = e^{-m_i}$, или *логистическую* функцию $\mathscr{L}(b, x_i) = \ln(1 + e^{-m_i})$, которая приводит к методу *логистической регрессии*, см. **??**. Тот же SVM эквивалентен применению кусочно-линейной функции потерь $\mathscr{L}(b, x_i) = (1 - m_i)_+$. Таким образом, принципы разделимости и минимизации эмпирического риска тесно связаны. Принцип разделимости легко связать также и с регуляризацией. Такие методы действительно разработаны, и обладают определённым комплексом преимуществ, например, *машины релевантных векторов*, RVM [24].

1.3.5 Принципы отделимости, закономерности, интерпретируемости

Принцип отделимости заключается в том, чтобы строить области в пространстве объектов X, каждая из которых отделяла бы объекты только какого-то одного из классов. Геометрическую форму этих областей предпочитают выбирать попроще: как правило, это шары, гиперплоскости или гиперпараллелепипеды. Поэтому эти области называют также *эталонами*.

В общем случае эталон класса $y \in Y$ — это предикат $\varphi_y \colon X \to \{0, 1\}$. Если $\varphi_y(x) = 1$, то говорят, что эталон φ_y *покрывает* объект x и относит его к классу y. Если $\varphi_y(x) = 0$, то считается, что эталон ничего не знает о классовой принадлежности объекта x, фактически, отказывается от его классификации.

В отличие от принципа разделимости, здесь не ставится задача классифицировать всю выборку с помощью одной единственной поверхности. Вместо этого строится множество эталонов, и каждый отделяет лишь небольшую часть своего класса.

Существует несколько способов собрать алгоритм классификации из набора эталонов $\varphi_{y1}, \ldots, \varphi_{yT_y}$. Чаще других используется *принцип голосования*: объект x относится к тому классу, за который голосует наибольшее число эталонов:

$$a(x) = \arg\max_{y \in Y} \sum_{t=1}^{T_y} \varphi_{yt}(x).$$

Для построения отдельных эталонов применяется *принцип закономерности*. Пусть $p(\varphi_y)$ — число *позитивных* объектов $x_i \in X^\ell$, правильно покрываемых эталоном φ_y, то есть для которых $\varphi_y(x_i) = 1$ и $y = y_i$. Соответственно, $n(\varphi_y)$ — число ошибочно покрываемых или *негативных* объектов, для которых $\varphi_y(x_i) = 1$, но $y \neq y_i$. Эталон $\varphi_y(x)$ называется *закономерностью*, если он покрывает достаточно много объектов и при этом допускает достаточно мало ошибок [8]:

$$p(\varphi_y) \geqslant \alpha; \qquad n(\varphi_y) \leqslant \beta;$$

где параметры α и β выбираются из априорных соображений, в зависимости от особенностей конкретной задачи.

Для поиска закономерностей в обучающих данных обычно применяют переборные алгоритмы. Допустим, в качестве эталонов выбраны шары вида

$$\varphi(x) = \big[\rho(x, x_0) \leqslant r_0\big],$$

где x_0 — центр шара, r_0 — его радиус, $\rho(x, x_0)$ — метрика в пространстве объектов X. Для поиска закономерностей-шаров перебираются возможные положения центра — обычно это один из объектов выборки: $x_0 \in \big\{x_i \in X^\ell \mid y_i = y\big\}$, затем при каждом x_0 перебираются возможные значения радиуса. Имеет смысл рассматривать только такие значения радиусов, при которых шары делят выборку по-разному. Поэтому берут $r_0 \in \big\{\rho(x_i, x_0) \mid x_i \in X^\ell\big\}$. Общее количество шаров составляет $O(\ell^2)$. Среди них отбираются только те, которые удовлетворяют критериям закономерности. Наконец, из них выбирается минимальное количество шаров, в совокупности покрывающих все объекты выборки. Это известная *задача о покрытии* множества системой его подмножеств. Алгоритмы её решения описаны в учебниках по дискретной оптимизации, например, в [???].

Простота эталонов обеспечивает *интерпретируемость* алгоритма классификации, что очень важно во многих приложениях. Например, в случае шаров можно так объяснять предлагаемые алгоритмом решения: «алгоритм $a(x)$ отнёс прецедент x к классу y потому, что он достаточно близок к прецеденту x_0 — типичному представителю класса y». Прецедентная логика алгоритма хорошо понятна экспертам в таких областях, как медицина, биометрия, страхование рисков, криминалистика.

Другой пример хорошо интерпретируемых закономерностей — логические правила, имеющие форму конъюнкции простых условий. Обычно «простые условия» — это сравнения значений признаков с порогами (хотя возможны и другие варианты):

$$\varphi_y(x) = \bigwedge_{j \in \omega} \big[f_j(x) \leqslant d_j\big],$$

где $\big(f_1(x), \ldots, f_n(x)\big)$ — признаковое описание объекта x. Подмножество признаков $\omega \subseteq \{1, \ldots, n\}$ и пороговые значения d_j являются параметрами правила φ_y, которые также подбираются путём перебора. Построенный алгоритм способен объяснять свои решения с помощью логических суждений. Например, «клиент x является хорошим заёмщиком, так как он менеджер среднего звена с жалованием более 2000 в месяц, имеет высшее образование и указал в анкете свой домашний телефон; 95% таких заёмщиков возвращают кредит в срок».

Особую ценность представляют подмножества ω, состоящие из небольшого числа признаков. Закономерности, образованные короткими фрагментами признакового

описания, называют *информативными наборами* признаков или *синдромами*. Синдромы записываются на естественном языке в виде простых и понятных правил «если–то», поэтому их можно рассматривать как знания о предметной области, извлечённые непосредственно из данных. Сам термин «синдром» произошёл от медицинских приложений распознавания образов.

Алгоритмы вычисления оценок (ABO) объединяют принципы близости, закономерности, и информативности в рамках единой модели, см. **??**.

Как было видно из приведённых примеров, алгоритмы, основанные на закономерностях, позволяют также оценивать степень уверенности (возможный процент ошибок) классификации, что позволяет применять их для оценки рисков.

1.3.6 Принципы самоорганизации и селекции моделей

Большой проблемой является выбор модели алгоритмов A, и далеко не всегда этот выбор обоснован какими-либо содержательными соображениями. Часто применяется линейный классификатор или линейная регрессия — просто потому, что их легче строить, и соответствующий метод обучения находится «под рукой».

Принцип самоорганизации моделей предполагает, что структура модели $a(x, \gamma)$ не известна заранее и выбирается из некоторого множества альтернатив, настолько богатого, что с его помощью можно описать практически любую зависимость. Метод обучения на основе самоорганизации решает две совершенно разные задачи — выбирает структуру модели и настраивает вектор параметров γ в выбранной модели. Первая задача решается путём направленного перебора большого числа моделей, при этом лучшая модель выбирается по *внешнему критерию*. Вторая задача решается путём оптимизации так называемых *внутренних критериев*. Принципы самоорганизации моделей, внешних и внутренних критериев были предложены А. Г. Ивахненко ещё в 1968 г. и легли в основу широко известного *метода группового учёта аргументов, МГУА* (group method of data handling, GMDH) [7].

Поясним различие между внутренними и внешними критериями на примере двух простейших частных случаев самоорганизации.

Задача *выбора модели* (model selection) заключается в следующем. Имеется множество альтернативных методов обучения $\mu_1, \ldots, \mu_T$. Каждый метод применяется к обучающей выборке, в результате строится множество алгоритмов $a_t = \mu_t(X^\ell)$, $t = 1, \ldots, T$. Возникает вопрос: какой алгоритм лучше? Оставить алгоритм с наименьшим значением эмпирического риска $Q_t = Q(a_t, X^\ell)$ было бы неверно, так как значение Q_t является заниженной оценкой ожидаемого риска. Это связано с явлением переобучения. Возможны ситуации, когда все значения Q_t одинаковы и равны нулю, тогда выбрать лучший алгоритм вообще не удастся. Функционал Q_t называется *внутренним критерием*, так как он используется в рамках конкретной модели для настройки её параметров. Для выбора лучшей из T моделей внутренний критерий не годится. Необходимо привлекать внешние данные, которые не были задействованы в процессе обучения. В простейшем случае исходная выборка разбивается на две части: обучающую X^ℓ и контрольную X^k, после чего лучшая модель выбирается по *внешнему критерию* минимума средней ошибки на контрольных данных:

$$t^* = \arg\min_{t=1,\ldots,T} Q(\mu_t(X^\ell), X^k).$$

Более сложный пример самоорганизации — задача *выбора информативных признаков* (features selection). На этапе формирования исходных данных, как правило, неизвестно, какие признаки, и в каком сочетании окажутся наиболее важными. Поэтому в признаковые описания включаются все данные об объектах, которые только доступны, и задача объективного выделения наиболее значимой части информации возлагается на алгоритм обучения. К сожалению, задача поиска информативных наборов признаков является NP-полной, то есть в общем случае требует полного перебора 2^n вариантов, где n — число признаков. На практике применяются эвристические схемы перебора, которые будут рассмотрены в **??**. В некоторых методах, таких как шаговая регрессия или МГУА, формируется только один информативный набор. Другие методы используют *принцип голосования*: строится большое количество информативных наборов, и окончательный ответ получается путём усреднения их ответов. Такая стратегия применяется в некоторых логических алгоритмах классификации и алгоритмах вычисления оценок (ABO), см. главу **??**. Но это уже относится к теме следующего раздела — принципу композиции.

1.3.7 Принцип композиции

При решении сложных задач классификации, регрессии и прогнозирования часто возникает следующая ситуация. Одна за другой предпринимаются попытки построить алгоритм, восстанавливающий искомую зависимость, однако качество всех построенных алгоритмов оставляет желать лучшего. В таких случаях имеет смысл объединить несколько алгоритмов в композицию, в надежде на то, что погрешности этих алгоритмов взаимно скомпенсируются.

В простейшем случае *алгоритмической композицией*, составленной из *базовых алгоритмов* $a_1, \ldots, a_T \colon X \to Y$ и *корректирующей операции* $F \colon Y^T \to Y$ называется алгоритм вида $a(x) = F(a_1(x), \ldots, a_T(x)),\ x \in X$.

Выделяются два основных принципа построения алгоритмических композиций — *усреднение* и *специализация*.

Простейшим примером усреднения является среднее арифметическое. Более общий случай — взвешенное среднее:

$$a(x) = \sum_{t=1}^{T} w_t a_t(x), \qquad \sum_{t=1}^{T} w_t = 1, \qquad x \in X,$$

где w_t — весовые коэффициенты. Обычно предполагается, что вес базовых алгоритмов неотрицателен, и что вес w_t тем больше, чем выше качество алгоритма a_t. Для настройки весов можно применять стандартные линейные методы классификации и регрессии, рассматривая векторы $\big(a_1(x), \ldots, a_T(x)\big)$ как признаковые описания объектов $x \in X$. Существуют и специализированные методы настройки весов в линейных композициях, например, метод бустинга, подробно разбираемый в **??**.

Принцип усреднения не ограничивается линейными композициями. Например, в **??** рассматриваются нелинейные монотонные корректирующие операции. Определяющей чертой, отличающей усреднение от специализации, является то, что корректирующая операция не знает, в какой области пространства находится объект, и может лишь комбинировать ответы, полученные от базовых алгоритмов.

Согласно *принципу специализации* пространство объектов делится на области, в каждой из которых настраивается свой алгоритм. Исходная задача разбивается

на более простые подзадачи по принципу «разделяй и властвуй». Формально алгоритм представляется также в виде линейной комбинации, однако теперь весовые коэффициенты w_t не постоянны, а зависят от положения объекта в пространстве X, и называются *функциями компетентности*.

$$a(x) = \sum_{t=1}^{T} w_t(x) a_t(x), \qquad \sum_{t=1}^{T} w_t(x) = 1, \qquad x \in X,$$

Здесь предполагается, что $w_t \colon X \to [0,1]$. Чем больше значение функции компетентности на объекте x, тем больше вклад алгоритма a_t в результат композиции. Если функция w_t принимает только два значения $\{0,1\}$, то множество $\{x \mid w_t(x) = 1\}$ называется *областью компетентности* базового алгоритма a_t. В общем случае функция w_t описывает область компетентности как нечёткое множество. Методы построения таких композиций, называемых *смесями экспертов* (mixture of experts), рассматриваются в **??**.

На принципе композиции основаны методы алгебраического подхода, взвешенное голосование, бустинг и бэггинг, метод комитетов, нейронные сети.

§1.4 О методологии интеллектуального анализа данных

Data Mining — это современная концепция анализа данных, изначально предполагающая, что данные могут быть неточными, разнородными, содержать пропуски, и при этом иметь гигантские объёмы. Необходимость в регулярном анализе таких данных возникла в результате повсеместного распространения информационных технологий, позволяющих детально протоколировать процессы бизнеса и производства.

Буквально «data mining» переводится как добыча или раскопка данных. Более-менее устоявшийся русский термин — *интеллектуальный анализ данных*.

Согласно общепринятому определению *Data Mining — это процесс обнаружения в сырых данных ранее неизвестных, нетривиальных, практически полезных, доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности.*

На самом деле, по составу решаемых задач data mining практически не отличается от стандартного набора средств, применяемых уже более полувека в области статистического анализа данных, поиска закономерностей и обучения по прецедентам. Основное различие заключается в эффективности алгоритмов и технологичности их применения. Подавляющее большинство классических процедур имеют квадратичное или даже кубическое (по числу объектов) время выполнения. При количестве объектов, превосходящем несколько десятков тысяч, они работают неприемлемо медленно даже на самых современных компьютерах. Специализированные алгоритмы data mining способны выполнять те же задачи за линейное или даже логарифмическое время без существенной потери точности.

1.4.1 Свойства реальных данных

В прикладных задачах исходные данные отражают сложность и разнообразие реальных процессов и явлений. Поэтому данные могут обладать рядом неприятных свойств, усложняющих поиск решения, причём эти свойства сочетаются друг с другом практически в любых комбинациях. Большое разнообразие методов обучения

по прецедентам во многом объясняется именно тем, что для каждого из этих случаев приходится искать свои подходы.

- *Неточность данных.* Значения признаков $f_j(x_i)$ и целевой переменной y_i могут измеряться с погрешностями. В некоторых случаях возможны грубые ошибки, приводящие к появлению редких, но больших отклонений — *выбросов* (outliers). Опасность зашумлённых данных в том, что обучаемые алгоритмы могут настраиваться на восстановление не только целевой зависимости, но и шума. В первую очередь, это относится к сложным моделям с большим числом свободных параметров.

  Для обеспечения помехоустойчивости методов обучения применяют оптимизацию сложности модели по критерию *скользящего контроля.* Для корректной обработки выбросов применяют предварительную фильтрацию данных или робастные методы оценивания параметров модели.

- *Неполнота данных.* Значения признаков $f_j(x_i)$ могут вообще отсутствовать, в этом случае говорят, что в данных имеются *пропуски* (missing data). Например, респондент может отказаться от ответа на некоторые пункты анкеты. В медицинских задачах пропуски являются скорее правилом, чем исключением — вряд ли стоит рассчитывать, что у каждого пациента будут проверены все возможные симптомы, применены все возможные виды обследований и лечения.

  Многие алгоритмы вообще не могут работать с пропущенными данными. В таких случаях применяют предварительную обработку — заполняют пропуски «спрогнозированными» значениями. Идея заключается в том, чтобы для каждого признака, имеющего пропущенные значения, решить вспомогательную задачу обучения по прецедентам. В качестве обучающей выборки берутся все объекты, для которых значение данного признака известно. Затем строится алгоритм, восстанавливающий зависимость данного признака от остальных. Этот алгоритм применяется к оставшимся объектам для заполнения пропусков.

  Другой подход — использовать алгоритмы, которые умеют игнорировать отдельные пропуски, не теряя при этом всей остальной информации. Например, таким свойством обладают логические алгоритмы классификации.

- *Противоречивость данных.* Прецеденты i и k могут противоречить друг другу. Например, $x_i = x_k$, но $y_i \neq y_k$. Они также могут противоречить априорным ограничениям. Например, может быть заранее известно, что зависимость y^* является монотонной, но $(x_i < x_k) \wedge (y_i > y_k)$ для некоторой пары прецедентов (i, k). Противоречивость может быть следствием неточности данных.

  Для обеспечения непротиворечивости применяют предварительную фильтрацию, отсеивая или заменяя противоречивые данные.

- *Разнородность данных.* Признаки $f_1, \ldots, f_n$ могут иметь различные типы (измеряться в разных шкалах).

  Некоторые методы обучения требуют однородности пространства признаков и приводят к неадекватным решениям, если это требование нарушается. При наличии признаков разного типа данные также проходят предварительную

обработку. Например, в логических алгоритмах классификации все признаки в конечном итоге преобразуются к бинарному типу (такая обработка может выполняться «на лету» в самом алгоритме).

- *Сложная структура данных.* Данные могут быть представлены в более сложной форме, чем стандартная матрица объектов-признаков. Это могут быть изображения, сигналы, тексты, графы, таблицы базы данных, и т. д.

  Для *извлечения признаков* применяют различные методы предварительной обработки данных. В задачах распознавания изображений и речи, классификации текстовых документов извлечение признаков является существенно более трудным этапом, в значительной степени предопределяющим успех распознавания.

  В некоторых задачах построение признаковых описаний оказывается вообще нецелесообразным, и объекты непосредственно сравниваются друг с другом. Тогда говорят о *беспризнаковом распознавании*. Например, сходство подписей или контурных изображений предметов можно оценивать как работу, необходимую для преобразования одного контура в другой, если представлять их в виде пластичных проволок. Аналогичным образом оценивается сходство сложных полимерных молекул, например, белков.

- *Недостаточность данных* (проблема малых выборок). Объектов может оказаться существенно меньше, чем признаков. В этом случае многие классические методы статистики и аппроксимации функций становятся неустойчивыми или вообще неприменимыми.

  Приходится прибегать к более изощрённым техникам: упрощать модель, оставляя только самые информативные признаки; искать закономерности, образованные короткими наборами признаков; привлекать дополнительную информацию в не-прецедентной форме.

- *Избыточность данных* (проблема сверхбольших выборок). В системах с автоматическим сбором и накоплением данных возникает противоположная проблема: данных настолько много, что обычные методы обрабатывают их крайне медленно.

  В зависимости от целей анализа могут применяться различные методы фильтрации или агрегирования данных. Например, эффективные субквадратичные алгоритмы кластеризации способны быстро выделить в массе исторических данных группу ситуаций, схожих с текущей наблюдаемой ситуацией.

1.4.2 Гипотеза представительности

Обучающую выборку называют *представительной*, если содержащейся в ней информации достаточно для восстановления зависимости с приемлемым качеством. Считается, что без этой гипотезы прогнозирование не имеет под собой научной основы и ничем не отличается от простого гадания.

К сожалению, оценить представительность выборки заранее, ещё до решения задачи, практически невозможно. Можно только утверждать, что если обученный

алгоритм хорошо работает на новых данных, то обучающая выборка представительна. Если же качество алгоритма не удовлетворяет, значит, либо выборка не представительна, либо выбранная модель алгоритмов не адекватна, либо метод обучения не позволяет найти в рамках данной модели хороший алгоритм.

На практике представительность выборки оценивают эмпирическим путём. В распоряжении исследователей имеется богатый арсенал методов обучения по прецедентам, применение которых в каждой конкретной задаче является отчасти делом техники, отчасти искусством, и требует известной изобретательности. Если все доступные средства самым добросовестным образом испробованы, но восстановить зависимость так и не удалось, значит выборка, скорее всего, была не представительна.

1.4.3 Классическое и информационное моделирование

Особенностью многих прикладных задач является отсутствие адекватной математической модели. В отдельных случаях модель может быть даже известна, но настолько сложна, что ни адаптировать, ни просчитать её за разумное время не представляется возможным. Эта ситуация возникает регулярно в таких трудно формализуемых областях, как медицина, геология, психология, социология, экономика. Даже в сложных технических системах создание и/или применение классической физической модели зачастую оказывается невозможным, несмотря на полное знание внутреннего устройства системы.

Спасает то, что во многих прикладных задач от построения детальной «физической» модели вполне можно оказаться. Обычно заказчика интересует не всестороннее изучение системы или явления, а регулярное решение небольшого круга задач, связанных с прогнозированием и принятием управленческих решений. Но тогда имеет смысл моделировать не саму систему, а лишь некоторые её информационные проявления. Сложность окружающей действительности компенсируется ограниченностью наших возможных действий.

Информационная модель, в отличие от физической, основывается не столько на экспертных знаниях о предметной области, сколько на общих принципах преобразования информации. Такие модели называют также *эвристическими*, поскольку они конструируются в значительной степени исходя из соображений здравого смысла, зачастую без строгого «физического» обоснования.

Вообще, процесс построения математических моделей в прикладных задачах можно разделить на два этапа.

Первый этап — формализация экспертных знаний о предметной области, в результате которой формируется структура модели. При построении физических моделей этот этап наиболее важен. В хорошей физической модели остаётся, как правило, небольшое число свободных параметров, имеющих чёткую содержательную интерпретацию. Эти модели узко специализированы и имеют фиксированные границы применимости.

Второй этап — настройка (идентификация) параметров модели по эмпирическим (экспериментальным) данным. Он существенно более важен для информационных моделей, когда данные являются едва ли не единственным, на что можно опереться. Информационные модели имеют значительно больше свободных параметров, зачастую не поддающихся содержательной интерпретации. Поэтому информационные модели существенно более универсальны. Одна и та же модель, скажем, бинарное

решающие дерево или нейронная сеть, может быть использована для классификации заёмщиков банка и диагностики раковой болезни. Среди информационных моделей немало «чёрных ящиков», которые способны неплохо решать практические задачи, но внутренняя логика этих решений остаётся загадкой даже для экспертов в данной прикладной области.

Чёткого различия между физическими и информационными моделями нет. Чем больше знаний о предметной области удаётся привлечь на первом этапе построения модели, тем более она физична.

1.4.4 Общая схема решения прикладных задач

Построение обучаемых алгоритмов в прикладных задачах — это исследовательская работа. Здесь нет универсальных рецептов, и решающую роль играют эксперименты на реальных данных. Следующая схема описывает наиболее типичный цикл исследований. Выполнение каждого этапа может опровергнуть очередную гипотезу и вернуть исследователя к любому из предыдущих этапов. Процесс поиска решения предполагает постепенное углубление в суть задачи.

- *Постановка задачи.* На этом этапе заказчик и разработчик находят общий язык — определяют цели решения задачи, договариваются о составе данных, фиксируют критерии качества решения.

- *Сбор данных.* Данный этап может меняться местами с предыдущим. Нередки ситуации, когда задачи анализа данных ставятся после того, как данные собраны. Но лучше, если процессы измерения и накопления данных заранее согласовываются с целями и методами последующего анализа.

- *Разведочный анализ, визуализация, формулировка гипотез.* На этом этапе выявляются первые «поверхностные» особенности решаемой задачи, выдвигаются и проверяются различные гипотезы о свойствах данных и стоящих за ними явлений, вырабатывается стратегия анализа данных. Принимаются решения о том, какие модели алгоритмов и методы настройки будут использоваться, в какой последовательности, какая для этого потребуется предварительная обработка данных, как будет оцениваться качество решений, какие промежуточные данные и в какой форме следует визуализировать, и т. д.

- *Предварительная обработка данных.* Каждый алгоритм предъявляет свои требования к исходным данным, например: нормированность, отсутствие пропусков, однородность (однотипность), линейная независимость, и т. д. Для приведения данных к требуемому виду применяются различные методы предварительной обработки. Если данные имеют слишком «сырую», сложную структуру, выполняется извлечение признаков или метрик. Возможно, при этом в признаковое описание будет включено большое количество избыточных, неинформативных признаков. Тогда на следующем этапе необходимо использовать специальные методы, способные эффективно отбирать информативные признаки.

- *Разработка моделей алгоритмов и методов их настройки.* Необходимость создания принципиально новых моделей и методов возникает довольно редко. За последние десятилетия их разработано огромное множество. В то же время,

некоторая адаптация готовой модели, как правило, позволяет повысить качество решения конкретной задачи.

- *Оценка качества алгоритмов.* Различаются два вида оценок: средняя ошибка на обучающей выборке и средняя ошибка на контрольных данных, не участвовавших в процессе настройки. Первая оценка практически всегда несколько занижена — это эффект переподгонки. Поэтому на практике качество алгоритмов принято оценивать по контрольным данным. Возникающая при этом проблема — как оставить побольше объектов для обучения, не уменьшая достоверности оценки — решается с помощью процедур скользящего контроля.

- *Отбор или коррекция алгоритмов.* Оценки качества могут использоваться для выбора лучшей модели алгоритмов. Более сильным приёмом считается комбинирование нескольких лучших алгоритмов. Существуют методы построения алгоритмических композиций, объединяющие этапы настройки, оценки качества и коррекции в рамках одной формальной процедуры.

- *Опытная эксплуатация* предполагает тестирование построенного алгоритма на новых данных. Эффект переподгонки может проявиться даже в тех случаях, когда качество алгоритма якобы подтверждается тестами на контрольных данных, но разработчик (явно или неявно) использовал результаты этих тестов для выбора наилучшего алгоритма. Только после успешного контроля качества на данных, совершенно незнакомых разработчику, принимается окончательное решение об использовании алгоритма.

- *Автоматизированное принятие решений* без участия человека-эксперта, как правило, является конечной целью построения обучаемых алгоритмов.

1.4.5 Методология тестирования обучаемых алгоритмов

Пока ещё не создан универсальный метод обучения по прецедентам, способный решать любые практические задачи одинаково хорошо. Каждый метод имеет свои границы применимости. У некоторых методов они более широкие, и с их помощью удаётся довольно успешно решать прикладные задачи из разных предметных областей. Другие методы более специализированы, и в среднем работают посредственно, но на узком классе задач дают наилучшие результаты. В общем случае довольно трудно понять заранее, какой метод окажется более успешным в конкретной задаче. Чаще всего приходится выбирать наилучший метод из имеющегося арсенала чисто эмпирически, по результатам численных экспериментов.

- *Эксперименты на модельных данных* используются, главным образом, на стадии отладки алгоритмов. Генерируются модельные выборки данных, как правило, в двумерном пространстве, чтобы работу алгоритма можно было наглядно представить на плоских графиках. Генерация данных выполняется либо с помощью датчика случайных чисел по заданным вероятностным распределениям, либо вообще детерминированным образом.

- *Эксперименты на реальных данных.* Обычно один или несколько наборов реальных данных имеются у прикладников, работающих с конкретной задачей

Задача	Источник	Объектов	Классов	Признаков				Пропусков
				бинар.	номин.	числ.	всего	
`iris`	UCI	150	3	0	0	4	4	нет
`german`	UCI	1000	2	2	11	7	20	нет

Таблица 1. Характеристики реальных задач классификации, используемых в качестве демонстрационных примеров.

в интересах конкретного заказчика. Но как быть, если речь идёт о разработке алгоритмов, предназначенных для решения широкого класса задач? Специально для тестирования и сравнения алгоритмов создаются репозитории прикладных задач. Один из наиболее известных — репозиторий задач классификации `UCI`[4], собранный в университете Ирвина (Калифорния, США). Он содержит около сотни задач [13].

За последние несколько десятков лет в машинном обучении сложился стандарт де факто: каждый новый метод тестируется на представительном наборе из нескольких десятков задач, в равных условиях с другими методами, чтобы показать, в каких случаях он превосходит своих предшественников.

1.4.6 Примеры реальных данных

В этом курсе для демонстрации и сравнения алгоритмов будут использоваться реальные данные. В Таблице 1 сведены общие характеристики этих задач. Сейчас на их примере мы покажем некоторые приёмы предварительного анализа и визуализации исходных данных.

`Iris.` Эта знаменитая задача классификации, на которой Р. А. Фишер в 1936 продемонстрировал работу своего *линейного дискриминанта* [15]. Объектами являются описания цветков ириса, классы — это три разновидности ирисов. Объекты описываются четырьмя количественными признаками: длина и ширина чашелистиков, длина и ширина лепестков. На Рис. 4 показаны всевозможные графики в осях двух признаков. Обычно такие графики строят на стадии предварительного обследования данных, если число признаков относительно невелико. В задаче с ирисами оказалось, что один из классов надёжно отделяется по 3-му, либо по 4-му признаку. Два других класса линейно не разделимы, но граница между ними чётко прослеживается на некоторых графиках. Вообще, задача `iris` считается очень простой. В других задачах простейшие двумерные проекции данных не столь информативны.

`German.` Задача классификации заёмщиков в одном из банков Германии на два класса: хороших — вернувших кредит, и плохих — не вернувших, вернувших не полностью или с просрочками. Это типичная задача с разнотипными признаками. Числовые признаки: размер кредита, срок кредита, процент по кредиту, срок проживания на одном месте, возраст, число уже взятых кредитов, число поручителей. Номинальные признаки: наличие задолженности, наличие оплаченных/задолженных кредитов в этом/другом банке, цель кредита (авто, ремонт, образование, всего 11 целей), срок

[4] http://www.ics.uci.edu/~mlearn/MLRepository.html.

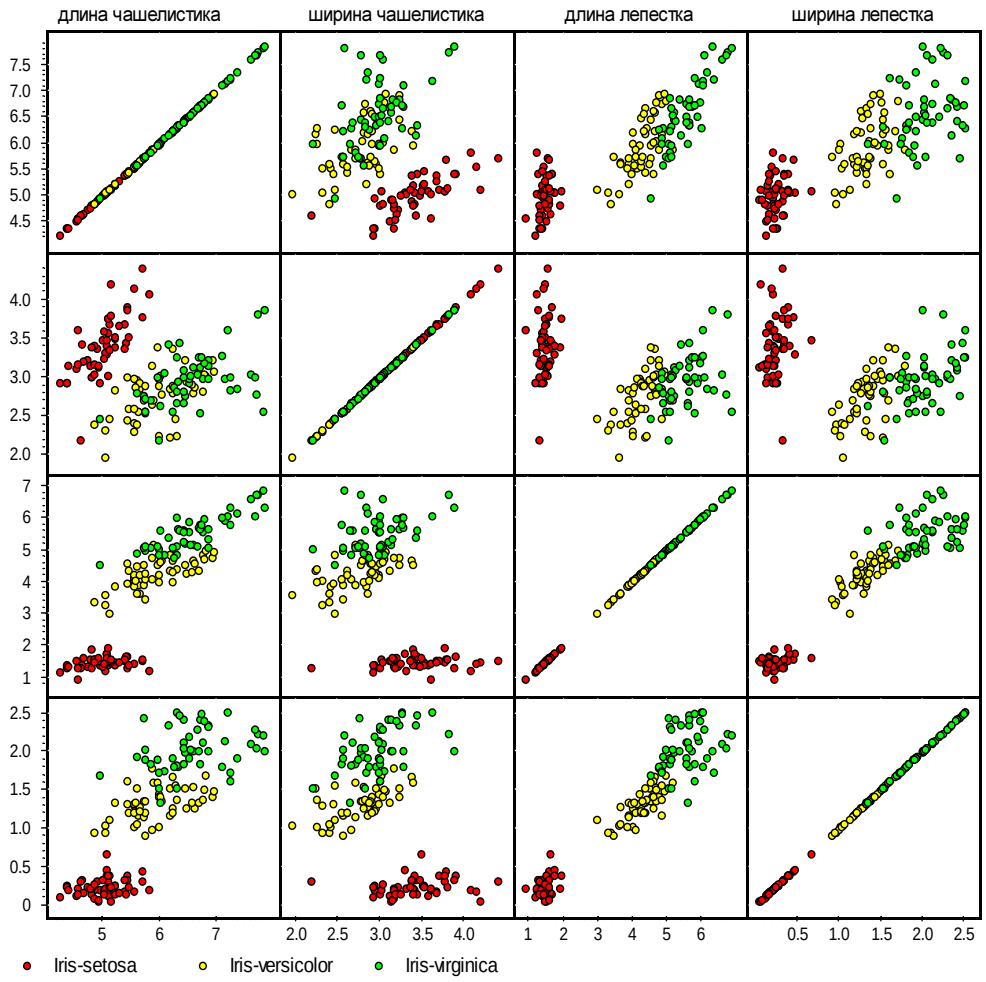

Рис. 4. Задача Фишера по классификации ирисов. Представление исходной выборки данных на n^2 двумерных графиках в осях «признак–признак», число признаков $n = 4$.

работы на одном месте, пол и семейное положение, образование, наличие телефона, и другие. Классы представлены в выборке не поровну: 700 хороших, 300 плохих. Цена ошибки I рода (принять плохого за хорошего) в 5 раз выше цены ошибки II рода (отвергнуть хорошего как плохого). Задача считается достаточно трудной, частота ошибок у лучших алгоритмов — около 25%.

1.4.7 Генерация модельных данных

Данный раздел носит справочный характер. В нём перечислены основные сведения, необходимые для генерации модельных выборок данных.

Моделирование случайных данных. Следующие утверждения позволяют генерировать случайные выборки с заданными распределениями [5]. Будем предполагать, что имеется стандартный способ получать равномерно распределённые на отрезке $[0, 1]$ случайные величины.

Утв. 1. Если случайная величина r равномерно распределена на $[0,1]$, то случайная величина $\xi = [r < p]$ принимает значение 1 с вероятностью p и значение 0 с вероятностью $1-p$.

Утв. 2. Если случайная величина r равномерно распределена на $[0,1]$, и задана возрастающая последовательность $F_0 = 0, F_1, \ldots, F_{k-1}, F_k = 1$, то дискретная случайная величина ξ, определяемая условием $F_{\xi-1} \leqslant r < F_\xi$, принимает значения $j = 1, \ldots, k$ с вероятностями $p_j = F_j - F_{j-1}$.

Утв. 3. Если случайная величина r равномерно распределена на $[0,1]$, и задана возрастающая на $\mathbb{R}$ функция $F(x)$, $0 \leqslant F(x) \leqslant 1$, то случайная величина $\xi = F^{-1}(r)$ имеет непрерывную функцию распределения $F(x)$.

Утв. 4. Если r_1, r_2 — две независимые случайные величины, равномерно распределённые на $[0,1]$, то *преобразование Бокса-Мюллера*

$$\xi_1 = \sqrt{-2\ln r_1}\sin 2\pi r_2;$$
$$\xi_2 = \sqrt{-2\ln r_1}\cos 2\pi r_2;$$

даёт две независимые нормальные случайные величины с нулевым матожиданием и единичной дисперсией: $\xi_1, \xi_2 \in \mathcal{N}(0,1)$.

Утв. 5. Если ξ — нормальная случайная величина из $\mathcal{N}(0,1)$, то случайная величина $\eta = \mu + \sigma\xi$ имеет нормальное распределение $\mathcal{N}(\mu, \sigma^2)$ с матожиданием μ и дисперсией σ^2.

Утв. 6. Пусть $x = (\xi_1, \ldots, \xi_n)$ — n-мерный вектор, составленный из независимых одномерных нормальных случайных величин ξ_i из $\mathcal{N}(0,1)$. Пусть V — невырожденная $n \times n$-матрица, $\mu \in \mathbb{R}^n$. Тогда вектор $x' = \mu + V^\intercal x$ имеет многомерное нормальное случайное распределение $\mathcal{N}(\mu, \Sigma)$ с вектором матожидания μ и ковариационной матрицей $\Sigma = V^\intercal V$.

Утв. 7. Пусть на вероятностном пространстве X заданы k плотностей распределения $p_1(x), \ldots, p_k(x)$. Пусть дискретная случайная величина ξ принимает значения $1, \ldots, k$ с вероятностями $w_1, \ldots, w_k$. Тогда случайный элемент $x \in X$, полученный согласно распределению $p_\xi(x)$, подчиняется *смеси распределений* $p(x) = \sum_{j=1}^k w_j p_j(x)$. На практике часто используют смеси многомерных нормальных распределений.

Утв. 8. Предыдущий случай обобщается на континуальные смеси распределений. Пусть на вероятностном пространстве X задано параметрическое семейство плотностей распределения $p(x,t)$, где $t \in \mathbb{R}$ — параметр. Пусть значение $\tau \in \mathbb{R}$ взято из распределения с плотностью $w(t)$. Тогда случайный элемент $x \in X$, полученный согласно распределению $p(x,\tau)$, подчиняется распределению $p(x) = \int_{-\infty}^{+\infty} w(t)p(x,t)\,dt$. Этот метод, называемый *методом суперпозиций*, позволяет моделировать широкий класс вероятностных распределений, представимых интегралом указанного вида.

Утв. 9. Пусть в $\mathbb{R}^n$ задана прямоугольная область $\Pi = [a_1, b_1] \times \ldots \times [a_n, b_n]$ и произвольное подмножество $G \subset \Pi$. Пусть $r = (r_1, \ldots, r_n)$ — вектор из n независимых

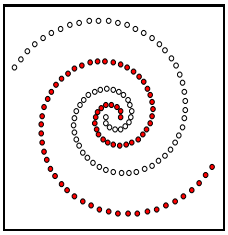

Рис. 5. Модельная выборка «спирали».

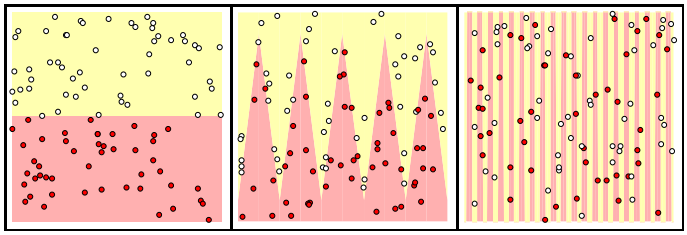

Рис. 6. Серия модельных выборок «пила».

случайных величин r_i, равномерно распределённых на $[a_i, b_i]$. *Метод исключения* состоит в том, чтобы генерировать случайный вектор r до тех пор, пока не выполнится условие $r \in G$. Тогда результирующий вектор r равномерно распределён на G. Этот метод вычислительно неэффективен, если объём G много меньше объёма П.

Неслучайные модельные данные позволяют наглядно продемонстрировать, в каких случаях одни методы работают лучше других. Один из классических примеров — две спирали на Рис. 5. Эта выборка хорошо классифицируется методом ближайших соседей, но непреодолимо трудна для линейных разделяющих правил. Если витки спиралей расположить ближе друг к другу, задача станет трудна и для метода ближайших соседей. Некоторые кусочно-линейные разделители справляются с задачей и в этом случае.

Обычно при создании модельных данных, как случайных, так и неслучайных, вводится параметр, плавно изменяющий задачу от предельно простой до предельно трудной. Это позволяет исследовать границы применимости метода. На Рис. 6 показана серия модельных задач классификации с двумя классами, обладающая таким свойством относительно метода ближайших соседей и некоторых других алгоритмов.

Резюме

1. Задача обучения по прецедентам: дана обучающая выборка, состоящая из пар «объект x_i, ответ y_i»; требуется построить алгоритм a, аппроксимирующий неизвестную целевую зависимость между объектами и ответами: $y = a(x)$. Решение задачи разделяется на две фазы. Первая, наиболее трудная — *обучение*: по выборке строится алгоритм a. Вторая — *применение*: построенный алгоритм a применяется для получения ответов.

2. *Метод минимизации эмпирического риска* является стандартным подходом к решению задач обучения по прецедентам. Фиксируется модель алгоритмов A и функционал средних потерь $Q(a, X^\ell)$. Обучение сводится к решению оптимизационной задачи: найти в заданной модели A алгоритм a, для которого величина средних потерь на заданной обучающей выборке X^ℓ минимальна.

3. Проблема *переобучения* заключается в том, что у построенного алгоритма a средние потери на новых объектах $Q(a, X^k)$ могут оказаться существенно выше, чем на обучающей выборке $Q(a, X^\ell)$. Оценивание *обобщающей способности* методов обучения является основной задачей статистической теории обучения.

Эмпирическое измерение обобщающей способности основано на методике *скользящего контроля*.

4. Стандартный способ представления обучающей выборки — матрица «объекты–признаки». *Признак* — это результат измерения какой-либо характеристики объекта, обычно числовой. Формально признак представляется как функция от объекта, и даже алгоритм может рассматриваться как признак.

5. Основные типы задач обучения по прецедентам: классификация, распознавание образов, принятие решений (множество ответов конечно), регрессия (ответы — действительные числа), прогнозирование (классификация или регрессия, когда ответы относятся к будущему времени), кластеризация (ответы не заданы, объекты классифицируются на основе сходства), анализ клиентских сред (рассматривается взаимодействие множества субъектов с множеством объектов).

6. В реальных задачах данные могут быть неполными, неточными, противоречивыми; признаки могут быть разнотипными, иметь сложную структуру или вообще отсутствовать; объём данных может быть как недостаточным, так и избыточным.

7. Процесс поиска алгоритма в типичном случае предполагает выполнение этапов разведочного анализа и предварительной обработки данных, затем оценивания, выбора и тестирования моделей алгоритмов. Некоторые из этих этапов могут повторяться многократно, пока не будет найдено приемлемое решение.

Список литературы

[1] *Вапник В. Н.* Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979.

[2] *Вапник В. Н., Червоненкис А. Я.* О равномерной сходимости частот появления событий к их вероятностям // *ДАН СССР*. — 1968. — Т. 181, № 4. — С. 781–784.

[3] *Вапник В. Н., Червоненкис А. Я.* О равномерной сходимости частот появления событий к их вероятностям // *Теория вероятностей и ее применения*. — 1971. — Т. 16, № 2. — С. 264–280.

[4] *Вапник В. Н., Червоненкис А. Я.* Теория распознавания образов. — М.: Наука, 1974.

[5] *Ермаков С. М., Михайлов Г. А.* Курс статистического моделирования. — М.: Наука, 1976.

[6] *Закс Ш.* Теория статистических выводов. — М.: Мир, 1975.

[7] *Ивахненко А. Г., Юрачковский Ю. П.* Моделирование сложных систем по экспериментальным данным. — М.: Радио и связь, 1987.

[8] *Лбов Г. С.* Методы обработки разнотипных экспериментальных данных. — Новосибирск: Наука, 1981.

[9] *Орлов А. И.* Эконометрика: Учебник для вузов. — М.: Экзамен, 2003. — С. 576.

[10] *Пытьев Ю. П.* Возможность. Элементы теории и применения. — М.: Эдиториал УРСС, 2000.

[11] *Тихонов А. Н., Арсенин В. Я.* Методы решения некорректных задач. — М.: Наука, 1986.

[12] *Трауб Д., Васильковскии Г., Вожьняковский Х.* Информация, неопределённость, сложность: Пер. с англ. — М.: Мир, 1988.

[13] *Asuncion A., Newman D.* UCI machine learning repository: Tech. rep.: University of California, Irvine, School of Information and Computer Sciences, 2007.
http://www.ics.uci.edu/~mlearn/MLRepository.html.

[14] *Boucheron S., Bousquet O., Lugosi G.* Theory of classification: A survey of some recent advances // *ESAIM: Probability and Statistics.* — 2005. — no. 9. — Pp. 323–375.
http://www.econ.upf.edu/~lugosi/esaimsurvey.pdf.

[15] *Fisher R. A.* The use of multiple measurements in taxonomic problem // *Ann. Eugen.* — 1936. — no. 7. — Pp. 179–188.

[16] *Garg A., Har-Peled S., Roth D.* On generalization bounds, projection profile, and margin distribution // ICML'02. — 2002.
http://citeseer.ist.psu.edu/garg02generalization.html.

[17] *Garg A., Roth D.* Margin distribution and learning algorithms // ICML'03. — 2003. — Pp. 210–217.
http://citeseer.ist.psu.edu/600544.html.

[18] *Herbrich R., Williamson R.* Algorithmic luckiness // *Journal of Machine Learning Research.* — 2002. — no. 3. — Pp. 175–212.
http://citeseer.ist.psu.edu/article/herbrich02algorithmic.html.

[19] *Kohavi R.* A study of cross-validation and bootstrap for accuracy estimation and model selection // 14th International Joint Conference on Artificial Intelligence, Palais de Congres Montreal, Quebec, Canada. — 1995. — Pp. 1137–1145.
http://citeseer.ist.psu.edu/kohavi95study.html.

[20] *Langford J.* Quantitatively tight sample complexity bounds. — 2002. — Carnegie Mellon Thesis.
http://citeseer.ist.psu.edu/langford02quantitatively.html.

[21] *Mason L., Bartlett P., Baxter J.* Direct optimization of margins improves generalization in combined classifiers // Proceedings of the 1998 conference on Advances in Neural Information Processing Systems II. — MIT Press, 1999. — Pp. 288–294.
http://citeseer.ist.psu.edu/mason98direct.html.

[22] *Rosset S., Zhu J., Hastie T.* Margin maximizing loss functions // Advances in Neural Information Processing Systems 16 / Ed. by S. Thrun, L. Saul, B. Schölkopf. — Cambridge, MA: MIT Press, 2004.
http://citeseer.ist.psu.edu/rosset03margin.html.

[23] *Rückert U., Kramer S.* Towards tight bounds for rule learning // Proc. 21th International Conference on Machine Learning, Banff, Canada. — 2004. — P. 90.
http://www.machinelearning.org/icml2004_proc.html.

[24] *Tipping M.* The relevance vector machine // Advances in Neural Information Processing Systems, San Mateo, CA. — Morgan Kaufmann, 2000.
http://citeseer.ist.psu.edu/tipping00relevance.html.