Вводная лекция

К. В. Воронцов

29 мая 2011 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 **2**
1.1 Основные понятия и определения . . . 2
1.1.1 Объекты и признаки . . . 2
1.1.2 Ответы и типы задач . . . 2
1.1.3 Модель алгоритмов и метод обучения . . . 3
1.1.4 Функционал качества . . . 3
1.1.5 Вероятностная постановка задачи обучения . . . 4
1.1.6 Проблема переобучения и понятие обобщающей способности . . . 6
1.2 Примеры прикладных задач . . . 7
1.2.1 Задачи классификации . . . 7
1.2.2 Задачи восстановления регрессии . . . 8
1.2.3 Задачи ранжирования . . . 9
1.2.4 Задачи кластеризации . . . 10
1.2.5 Задачи поиска ассоциаций . . . 11
1.2.6 Методология тестирования обучаемых алгоритмов . . . 12
1.2.7 Приёмы генерации модельных данных . . . 13

1 Введение: задачи обучения по прецедентам

В этой вводной лекции даются базовые понятия и обозначения, которые будут использоваться на протяжении всего курса. Приводятся общие постановки задач обучения по прецедентам и некоторые примеры прикладных задач.

§1.1 Основные понятия и определения

Задано множество *объектов* X, множество *допустимых 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 по выборке X^ℓ *восстановить зависимость* y^*, то есть построить *решающую функцию* (decision function) $a\colon X \to Y$, которая приближала бы целевую функцию $y^*(x)$, причём не только на объектах обучающей выборки, но и на всём множестве X. Решающая функция a должна допускать эффективную компьютерную реализацию; по этой причине будем её называть также *алгоритмом*.

1.1.1 Объекты и признаки

Признак (feature) f объекта x — это результат измерения некоторой характеристики объекта. Формально признаком называется отображение $f\colon X \to D_f$, где D_f — множество допустимых значений признака. В частности, любой алгоритм $a\colon X \to Y$ также можно рассматривать как признак.

В зависимости от природы множества D_f признаки делятся на несколько типов.

Если $D_f = \{0, 1\}$, то f — *бинарный* признак;

Если D_f — конечное множество, то f — *номинальный* признак;

Если D_f — конечное упорядоченное множество, то f — *порядковый* признак;

Если $D_f = \mathbb{R}$, то f — *количественный* признак.

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 размера $\ell \times n$, называют *матрицей объектов–признаков*:

$$F = \|f_j(x_i)\|_{\ell \times n} = \begin{pmatrix} f_1(x_1) & \ldots & f_n(x_1) \\ \ldots & \ldots & \ldots \\ f_1(x_\ell) & \ldots & f_n(x_\ell) \end{pmatrix}. \tag{1.1}$$

Матрица объектов–признаков является стандартным и наиболее распространённым способом представления исходных данных в прикладных задачах.

1.1.2 Ответы и типы задач

В зависимости от природы множества допустимых ответов Y задачи обучения по прецедентам делятся на следующие типы.

Если $Y = \{1, \ldots, M\}$, то это 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).

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

Если $Y = \mathbb{R}$, то это задача *восстановления регрессии* (regression estimation).

Задачи *прогнозирования* (forecasting) являются частными случаями классификации или восстановления регрессии, когда $x \in X$ — описание прошлого поведения объекта x, $y \in Y$ — описание некоторых характеристик его будущего поведения.

1.1.3 Модель алгоритмов и метод обучения

Опр. 1.1. *Моделью алгоритмов называется параметрическое семейство отображений $A = \{g(x, \theta) \mid \theta \in \Theta\}$, где $g : X \times \Theta \to Y$ — некоторая фиксированная функция, Θ — множество допустимых значений параметра θ, называемое пространством параметров или пространством поиска (search space).*

Процесс подбора оптимального параметра модели θ по обучающей выборке X^ℓ называют *настройкой* (fitting) или *обучением* (training, learning)[1] алгоритма $a \in A$.

Опр. 1.2. *Метод обучения (learning algorithm)*[2] *— это отображение $\mu : (X \times Y)^\ell \to A$, которое произвольной конечной выборке $X^\ell = (x_i, y_i)_{i=1}^\ell$ ставит в соответствие некоторый алгоритм $a \in A$. 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 $a = \mu(X^\ell)$.

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.

1.1.4 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.*

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

[2] В английской терминологии алгоритмы a называют классификаторами, гипотезами или просто функциями (единого термина нет), а методы обучения μ — алгоритмами обучения. По-русски говорят «метод ближайших соседей», «метод потенциальных функций», «метод опорных векторов», «метод обратного распространения ошибок», и т. д., имея в виду конкретные методы обучения по выборке. Поэтому мы употребляем термин «метод обучения», хотя это несколько противоречит сложившейся в мире терминологии («learning method» не говорят — это термин из педагогики).

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

$$Q(a, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathscr{L}(a, x_i). \tag{1.2}$$

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.

Наиболее часто используются следующие функции потерь, при $Y \subseteq \mathbb{R}$:

$\mathscr{L}(a, x) = [a(x) \neq y^*(x)]$ — индикатор ошибки, обычно применяется в задачах классификации;

$\mathscr{L}(a, x) = |a(x) - y^*(x)|$ — 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; обычно применяется в задачах регрессии.

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:

$$\mu(X^\ell) = \arg\min_{a \in A} Q(a, X^\ell). \tag{1.3}$$

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

1.1.5 Вероятностная постановка задачи обучения

В задачах обучения по прецедентам элементы множества X — это не реальные объекты, а лишь доступные данные о них. Данные могут быть *неточными*, поскольку измерения значений признаков $f_j(x)$ и целевой зависимости $y^*(x)$ обычно выполняются с погрешностями. Данные могут быть *неполными*, поскольку измеряются не все мыслимые признаки, а лишь физически доступные для измерения. В результате одному и тому же описанию x могут соответствовать различные объекты и различные ответы. В таком случае $y^*(x)$, строго говоря, не является функцией. Устранить эту некорректность позволяет *вероятностная постановка задачи*.

Вместо существования неизвестной целевой зависимости $y^*(x)$ предположим существование неизвестного вероятностного распределения на множестве $X \times Y$ с плотностью $p(x, y)$, из которого случайно и независимо выбираются ℓ наблюдений $X^\ell = (x_i, y_i)_{i=1}^\ell$. Такие выборки называются *простыми* или *случайными одинаково распределёнными* (independent identically distributed, i.i.d.).

Вероятностная постановка задачи считается более общей, 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.

Принцип максимума правдоподобия. При вероятностной постановке задачи вместо модели алгоритмов $g(x,\theta)$, аппроксимирующей неизвестную зависимость $y^*(x)$, задаётся модель совместной плотности распределения объектов и ответов $\varphi(x,y,\theta)$, аппроксимирующая неизвестную плотность $p(x,y)$. Затем определяется значение параметра θ, при котором выборка данных X^ℓ максимально правдоподобна, то есть наилучшим образом согласуется с моделью плотности.

Если наблюдения в выборке X^ℓ независимы, то совместная плотность распределения всех наблюдений $p(X^\ell)$ может быть представлена в виде произведения значений плотности $p(x,y)$ в каждом наблюдении:

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

Если подставить вместо $p(x,y)$ модель плотности $\varphi(x,y,\theta)$, то получится *функция правдоподобия* (likelihood):

$$L(\theta,X^\ell) = \prod_{i=1}^{\ell}\varphi(x_i,y_i,\theta).$$

Чем выше значение правдоподобия, тем лучше выборка согласуется с моделью. Значит, нужно искать значение параметра θ, при котором значение $L(\theta,X^\ell)$ максимально. В математической статистике это называется *принципом максимума правдоподобия*. Его формальные обоснования можно найти, например, в [6].

После того, как значение параметра θ найдено, искомый алгоритм $a(x)$ строится по плотности $\varphi(x,y,\theta)$ несложно.

Связь максимизации правдоподобия с минимизацией эмпирического риска. Вместо максимизации L удобнее минимизировать функционал $-\ln L$, поскольку он аддитивен (имеет вид суммы) по объектам выборки:

$$-\ln L(\theta,X^\ell) = -\sum_{i=1}^{\ell}\ln\varphi(x_i,y_i,\theta) \to \min_\theta. \tag{1.4}$$

Этот функционал совпадает с функционалом эмпирического риска (1.2), если определить *вероятностную функцию потерь* $\mathscr{L}(a_\theta,x) = -\ell\ln\varphi(x,y,\theta)$. Такое определение потери вполне естественно — чем хуже пара (x_i,y_i) согласуется с моделью φ, тем меньше значение плотности $\varphi(x_i,y_i,\theta)$ и выше величина потери $\mathscr{L}(a_\theta,x)$.

Покажем на примере задачи регрессии, что вероятностная и функциональная постановки тесно связаны.

Пример 1.1. Пусть задана модель $g(x,\theta)$. Примем дополнительное вероятностное предположение, что ошибки $\varepsilon(x,\theta) = g(x,\theta) - y^*(x)$ имеют нормальное распределение $\mathcal{N}(\varepsilon;0,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp\big(-\frac{\varepsilon^2}{2\sigma^2}\big)$ с нулевым средним и дисперсией σ^2. Тогда модель плотности имеет вид $\varphi(x,y,\theta) = p(x)\varphi(y\,|\,x,\theta) = p(x)\mathcal{N}\big(g(x,\theta)-y^*(x);0,\sigma^2\big)$. Отсюда следует, что вероятностная функция потерь совпадает с квадратичной с точностью до констант C_0 и C_1, не зависящих от параметра θ:

$$-\ln\varphi(x,y,\theta) = -\ln p(x)\mathcal{N}\big(g(x,\theta)-y^*(x);0,\sigma^2\big) = C_0 + C_1\big(g(x,\theta)-y^*(x)\big)^2.$$

Аналогично устанавливается эквивалентность минимизации эмпирического риска и максимизации правдоподобия и при других функциях потерь. Обе постановки приводят в итоге к схожим оптимизационным задачам.

1.1.6 Проблема переобучения и понятие обобщающей способности

Минимизацию эмпирического риска следует применять с известной долей осторожности. Если минимум функционала $Q(a, X^\ell)$ достигается на алгоритме a, то это ещё не гарантирует, что a 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.

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 материала обучения. Отсюда вывод: 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 одного и того же 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*

$$\mathsf{P}_{X^\ell, X^k}\big\{Q(\mu(X^\ell), X^k) > \varepsilon\big\} < \eta. \tag{1.5}$$

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 $1 - \eta$.

Получение оценок вида (1.5) является фундаментальной проблемой статистической теории обучения. Первые оценки были получены в конце 60-х годов В. Н. Вапником и А. Я. Червоненкисом [2, 3, 4]. В настоящее время статистическая теория развивается очень активно [8], однако для многих практически интересных случаев оценки обобщающей способности либо неизвестны, либо сильно завышены. Численно точные оценки получены лишь для некоторых частных случаев [11, 9, 12].

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):

$$\mathrm{CV}(\mu, X^L) = \frac{1}{N} \sum_{n=1}^{N} Q(\mu(X_n^\ell), X_n^k). \tag{1.6}$$

Возможны различные варианты скользящего контроля, отличающиеся способами разбиения выборки X^L [10]. В простейшем варианте разбиения генерируются случайным образом, число N берётся в диапазоне от 20 до 100. Стандартом «де факто» считается методика *$t\times q$-кратного скользящего контроля* ($t\times q$-fold cross-validation), когда выборка случайным образом разбивается на q блоков равной (или почти равной) длины, каждый блок по очереди становится контрольной выборкой, а объединение всех остальных блоков — обучающей. Выборка X^L по-разному t раз разбивается на q блоков. Итого получается $N = tq$ разбиений. Данная методика даёт более точные оценки за счёт равномерной представленности объектов в обучении и в контроле. Каждый объект оказывается контрольным ровно в t разбиениях из N.

Недостатками скользящего контроля являются: вычислительная неэффективность, высокая дисперсия, неполное использование имеющихся данных для обучения из-за сокращения длины обучающей выборки с L до ℓ.

§1.2 Примеры прикладных задач

Прикладные задачи классификации, регрессии и прогнозирования встречаются в самых разных областях человеческой деятельности, причём число приложений постоянно растёт.

1.2.1 Задачи классификации

Пример 1.2. 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 1.3. 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 1.4. 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 сегмент 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.

1.2.2 Задачи восстановления регрессии

Пример 1.5. 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 физических экспериментах, в промышленном производстве, в экономике.

Пример 1.6. 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 получать эти данные от кассовых аппаратов и накапливать в едином хранилище данных. Для увеличения точности прогнозов необходимо учитывать различные внешние факторы, влияющие на спрос: рекламные кампании, социально-демографические условия, активность конкурентов, праздники, и даже погодные условия. 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, чем от завышенного.

Пример 1.7. Задача *предсказания рейтингов* решается интернет-магазинами, особенно книжными, видео и аудио. Приобретая товар, клиент имеет возможность выставить ему рейтинг, например, 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 ранжированный список схожих товаров, пользующихся популярностью у схожих клиентов. Основная задача — прогнозировать рейтинги товаров, которые данный клиент ещё не приобретал. Роль матрицы объектов–признаков играет матрица клиентов–товаров, заполненная значениями рейтингов. Как правило, она сильно разрежена и имеет более 99% пустых ячеек. Фиксированного целевого признака в этой задаче нет. Алгоритм должен предсказывать рейтинги для любых незаполненных ячеек матрицы. Данный тип задач выделяют особо и называют задачами *коллаборативной фильтрации* (collaborative filtering).

О трудности и актуальности этой задачи говорит следующий факт. В октябре 2006 года крупнейшая американская компания Netflix, занимающаяся видеопрокатом через Internet, объявила международный конкурс с призом в 1 миллион долларов тому, кто сможет на 10% улучшить точность прогнозирования рейтингов, по сравнению с системой Netflix Cinematch (см. http://www.netflixprize.com). Примечательно, что прогнозы самой Cinematch были лишь на те же 10% точнее элементарных прогнозов по средним рейтингам фильмов. Компания крайне заинтересована в увеличении точности прогнозов, поскольку около 70% заказов поступают через рекомендующую систему. Конкурс успешно завершился только через два с половиной года.

1.2.3 Задачи ранжирования

Задачи ранжирования возникают во многих приложениях информационного поиска. Результатом поиска по запросу может оказаться настолько длинный список, что пользователь физически не сможет его просмотреть. Хотелось бы упорядочить

пункты списка по убыванию *релевантности* — степени их соответствия запросу. Критерий упорядочения в явном виде неизвестен, хотя человек легко отличает релевантную информацию от нерелевантной. Поэтому просят экспертов (их называют асессорами) разметить заранее сформированную обучающую выборку, и затем решают задачу обучения ранжированию (learning to rank).

Пример 1.8. Задача *ранжирования текстовых документов*, найденных в Интернете по запросу пользователя, решается всеми современными поисковыми машинами. Объектами являются пары «запрос, документ», ответами — оценки релевантности, сделанные асессорами. В зависимости от методологии формирования обучающей выборки оценки асессоров могут быть бинарными (релевантен, не релевантен) или порядковыми (релевантность в баллах). Признаками являются числовые характеристики, вычисляемые по паре «запрос, документ». Текстовые признаки основаны на подсчёте числа вхождений слов запроса в документы. Возможны многочисленные варианты: с учётом синонимов или без, с учётом числа вхождений или без, во всём документе или только в заголовках, и т. д. Ссылочные признаки основаны на подсчёте числа ссылок на данный документ и оценивают документ независимо от запроса. Кликовые признаки основаны на подсчёте числа обращений к данному документу.

Пример 1.9. Задачу *предсказания рейтингов* из примера 1.7 на практике лучше ставить как задачу ранжирования. Пользователю выдаётся список рекомендаций, поэтому важно обеспечить высокую релевантность относительно небольшого числа товаров, попадающих в вершину списка. Среднеквадратичная ошибка предсказания рейтингов, которую предлагалось минимизировать в условии конкурса Netflix, в данном случае не является адекватной мерой качества.

Заметим, что в этой задаче в роли асессоров выступают все пользователи рекомендательного сервиса. Покупая товар или выставляя оценку, пользователь пополняет обучающую выборку и тем самым способствует улучшению качества сервиса.

1.2.4 Задачи кластеризации

Задачи кластеризации (clustering) отличаются от классификации (classification) тем, что в них не задаются ответы $y_i = y^*(x_i)$. Известны только сами объекты x_i, и требуется разбить выборку на подмножества (кластеры) так, чтобы каждый кластер состоял из схожих объектов, а объекты разных кластеров существенно отличались. Для этого необходимо задавать функцию расстояния на множестве объектов. Число кластеров также может задаваться, но чаще требуется определить и его.

Пример 1.10. 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 респондентов, 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.

Задачи кластеризации, в которых часть объектов (как правило, незначительная) размечена по классам, называются задачами *с частичным обучением* (semi-supervised learning). Считается, что они не сводятся непосредственно к классификации или кластеризации, и для их решения нужны особые методы.

Пример 1.11. Задача *рубрикации текстов* возникает при работе с большими коллекциями текстовых документов. Допустим, имеется некоторый иерархический рубрикатор, разработанный экспертами для данной предметной области (например, для спортивных новостей), или для всех областей (например, универсальный десятичный классификатор УДК). Имеется множество документов, классифицированных по рубрикам вручную. Требуется классифицировать по тем же рубрикам второе множество документов, которое может быть существенно больше первого. Для решения данной задачи используется функция расстояния, сравнивающая тексты по составу терминов. Терминами, как правило, являются специальные понятия предметной области, собственные имена, географические названия, и т. д. Документы считаются схожими, если множества их терминов существенно пересекаются.

1.2.5 Задачи поиска ассоциаций

Задача *поиска ассоциативных правил* (association rule induction) вынесена в отдельный класс и относится к задачам обучения без учителя, хотя имеет много общего с задачей классификации.

Пример 1.12. Задача *анализа рыночных корзин* (market basket analysis) состоит в том, чтобы по данным о покупках товаров в супермаркете (буквально, по чекам) определить, какие товары часто совместно покупаются. Эта информация может быть полезной для оптимизации размещения товаров на полках, планирования рекламных кампаний (промо-акций), управления ассортиментом и ценами. В данной задаче объекты соответствуют чекам, признаки являются бинарными и соответствуют товарам. Единичное значение признака $f_j(x_i) = 1$ означает, что в i-м чеке зафиксирована покупка j-го товара. Задача состоит в том, чтобы выявить все наборы товаров, которые часто покупают вместе. Например, «если куплен хлеб, то с вероятностью 60% будет куплено и молоко». Известен пример, вошедший во все учебники по бизнес-аналитике, когда система поиска ассоциативных правил обнаружила неочевидную закономерность: вечером перед выходными днями сильно возрастают совместные продажи памперсов и пива. Размещение холодильников с дорогими сортами пива рядом с памперсами позволило сети супермаркетов очень быстро окупить внедрение системы анализа данных. Позже социологи предложили объяснение данному явлению, однако для бизнеса это уже не имело значения.

Пример 1.13. Задача *выделения терминов* (term extraction) из текстов, решаемая перед задачей рубрикации (см. пример 1.11), может быть сведена к поиску ассоциаций. Терминами считаются отдельные слова или устойчивые словосочетания, которые часто встречаются в небольшом подмножестве документов, и редко — во всех

остальных. Множество часто совместно встречающихся терминов образует тему, скорее всего, соответствующую некоторой рубрике.

1.2.6 Методология тестирования обучаемых алгоритмов

Пока ещё не создан универсальный метод обучения по прецедентам, способный решать любые практические задачи одинаково хорошо. Каждый метод имеет свои преимущества, недостатки и границы применимости. Некоторые методы предназначены для решения широкого класса задач и подходят для разных предметных областей. Другие методы более специализированы, в среднем работают посредственно, но на узком классе задач демонстрируют наилучшие результаты. На практике приходится проводить численные эксперименты, чтобы понять, какой метод из имеющегося арсенала лучше подходит для конкретной задачи. Обычно для этого методы сравниваются по скользящему контролю (1.6).

Существует два типа экспериментальных исследований, отличающихся целями и методикой проведения.

Эксперименты на модельных данных. Их цель — выявление границ применимости метода обучения; построение примеров удачной и неудачной его работы; понимание, на что влияют параметры метода обучения. Модельные эксперименты часто используются на стадии отладки метода. Модельные выборки сначала генерируются в двумерном пространстве, чтобы работу метода можно было наглядно представить на плоских графиках. Затем исследуется работа метода на многомерных данных, при различном числе признаков. Генерация данных выполняется либо с помощью датчика случайных чисел по заданным вероятностным распределениям, либо детерминированным образом. Часто генерируется не одна модельная задача, а целая серия, параметризованная таким образом, чтобы среди задач оказались как заведомо «лёгкие», так и заведомо «трудные»; при такой организации эксперимента точнее выявляются границы применимости метода.

Эксперименты на реальных данных. Их цель — либо решение конкретной прикладной задачи, либо выявление «слабых мест» и границ применимости конкретного метода. В первом случае фиксируется задача, и к ней применяются многие методы, или, возможно, один и тот же метод при различных значениях параметров. Во втором случае фиксируется метод, и с его помощью решается большое число задач (обычно несколько десятков). Специально для проведения таких экспериментов создаются общедоступные репозитории реальных данных. Наиболее известный — репозиторий UCI (университета Ирвина, Калифорния), доступный по адресу `http://archive.ics.uci.edu/ml`. Он содержит около двух сотен задач, в основном классификации, из самых разных предметных областей [7].

Полигон алгоритмов классификации. В научных статьях по машинному обучению принято приводить результаты тестирования предложенного нового метода обучения в сравнении с другими методами на представительном наборе задач. Сравнение должно производиться в равных условиях по одинаковой методике; если это скользящий контроль, то при одном и том же множестве разбиений. Несмотря на значительную стандартизацию таких экспериментов, результаты тестирования одних

и тех же методов на одних и тех же задачах, полученные разными авторами, всё же могут существенно различаться. Проблема в том, что используются различные реализации методов обучения и методик тестирования, а проведённый кем-то ранее эксперимент практически невозможно воспроизвести во всех деталях. Для решения этой проблемы разработан *Полигон алгоритмов классификации*, доступный по адресу http://poligon.MachineLearning.ru. В этой системе реализована унифицированная расширенная методика тестирования и централизованное хранилище задач. Реализация алгоритмов классификации, наоборот, децентрализована. Любой пользователь Интернет может объявить свой компьютер вычислительным сервером Полигона, реализующим один или несколько методов классификации. Все результаты тестирования сохраняются как готовые отчёты в базе данных системы и могут быть в любой момент выданы по запросу без проведения трудоёмких вычислений заново.

1.2.7 Приёмы генерации 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.

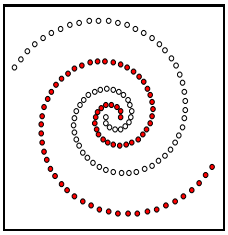

Рис. 1. Модельная выборка «спирали».

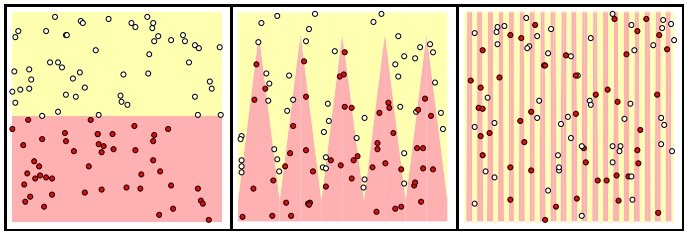

Рис. 2. Серия модельных выборок «пила».

Утв. 6. Пусть n-мерный вектор $x = (\xi_1, \ldots, \xi_n)$ составлен из независимых нормальных случайных величин $\xi_i \sim \mathcal{N}(0, 1)$. Пусть V — невырожденная $n \times n$-матрица, $\mu \in \mathbb{R}^n$. Тогда вектор $x' = \mu + V^\intercal x$ имеет многомерное нормальное 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 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 Π.

Неслучайные модельные данные позволяют наглядно продемонстрировать, в каких случаях одни методы работают лучше других.

Один из классических примеров — две спирали на Рис. 1. 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. 2 показана серия модельных задач классификации с двумя классами, обладающая таким свойством относительно метода ближайших соседей и некоторых других алгоритмов.

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] *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.

[8] *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.

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

[10] *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.

[11] *Langford J.* Quantitatively Tight Sample Complexity Bounds: Ph.D. thesis / Carnegie Mellon Thesis. — 2002.
http://citeseer.ist.psu.edu/langford02quantitatively.html.

[12] *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.