Определение 8.1 (расстояние Минковского). *Если* $X = \mathbb{R}^d$, *то* расстояние Минковского *порядка* $p > 0$ *определяется формулой*

$$\mathrm{Dis}_p(\mathbf{x},\mathbf{y}) = \left(\sum_{j-1}^{d} |x_j - y_j|^p\right)^{1/p} = \|\mathbf{x}-\mathbf{y}\|_p,$$

где $\|\mathbf{z}\|_p = (\sum_{j=1}^{d} |z_j|^p)^{1/p}$ *–* p*-норма* (*иногда обозначается* L_p) *вектора* $\mathbf{z}$. *Мы часто будем называть* Dis_p *просто* p*-нормой.*

Так, 2*-норма* – не что иное, как всем знакомое *евклидово расстояние*:

$$\mathrm{Dis}_2(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{j=1}^{d} |x_j - y_j|^2} = \sqrt{(\mathbf{x}-\mathbf{y})^{\mathrm{T}}(\mathbf{x}-\mathbf{y})},$$

которое измеряет расстояние «по прямой». Два других значения p можно соотнести с примером из мира шахмат. 1*-норма* иначе называется *манхэттенским расстоянием*, или *расстоянием городских кварталов*:

$$\mathrm{Dis}_1(\mathbf{x},\mathbf{y}) = \sum_{j-1}^{d} |x_j - y_j|.$$

Это расстояние, которое мы проходим, двигаясь только параллельно осям координат, – как при поездке на такси в Манхэттене или в других городах с продольно-поперечной дорожной сетью. И вместе с тем это расстояние в восприятии нашей воображаемой «короладьи». Если увеличивать p, то расстояние будет все в большей и большей степени определяться наибольшим расстоянием вдоль одной из осей, откуда можно сделать вывод, что $\mathrm{Dis}_\infty(\mathbf{x},\mathbf{y}) = \max_j |x_j - y_j|$. Это расстояние в смысле короля, который может двигаться в любом направлении, но только на одну клетку, его также называют *расстоянием Чебышева*. На рис. 8.3 слева показаны точки, равноудаленные от начала координат в смысле расстояний Минковского разного порядка. Легко видеть, что только евклидово расстояние инвариантно относительно поворота, то есть при любом $p \neq 2$ направления осей координат играют особую роль. В определении расстояния Минковского начало координат не фигурирует, поэтому при любом p оно инвариантно относительно параллельных переносов, но ни при каком p не является инвариантным относительно масштабирования.

Иногда можно встретить упоминание о 0*-норме* (или норме L_0), которая подсчитывает количество ненулевых элементов в векторе. Соответствующее ей расстояние – это количество позиций, в которых векторы $\mathbf{x}$ и $\mathbf{y}$ различаются. Строго говоря, это не расстояние Минковского, но его можно определить следующим образом:

$$\mathrm{Dis}_0(\mathbf{x},\mathbf{y}) = \sum_{j-1}^{d} (x_j - y_j)^0 = \sum_{j-1}^{d} I[x_j = y_j],$$

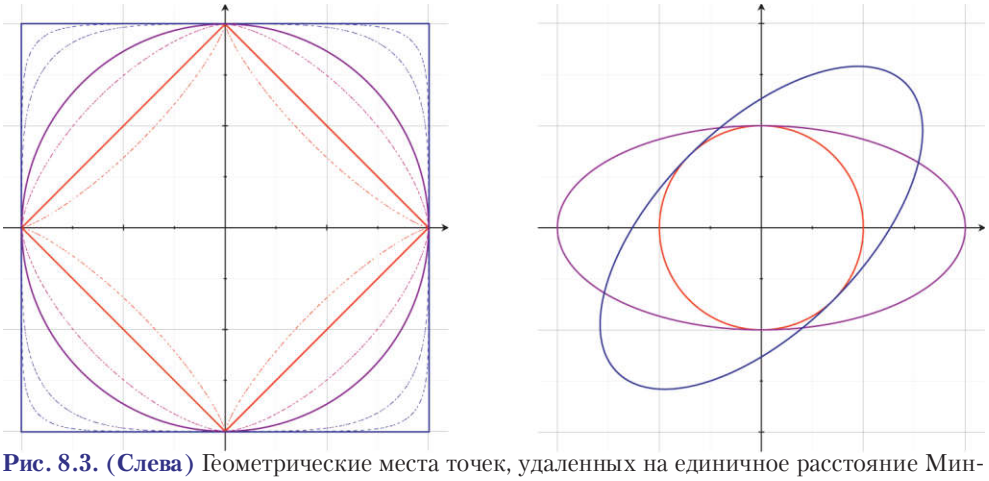

Рис. 8.3. (Слева) Геометрические места точек, удаленных на единичное расстояние Минковского порядка p от начала координат, при (изнутри наружу) $p = 0.8$, p = 1 (манхэттенское расстояние, повернутый **красный** квадрат), $p = 1.5$, $p = 2$ (евклидово расстояние, **фиолетовая** окружность), $p = 4$, $p = 8$ и $p = \infty$ (расстояние Чебышева, **синий** прямоугольник). Отметим, что в точках на осях координат все расстояния совпадают, но в остальном с ростом p фигура «разбухает». Однако условию инвариантности относительно поворота удовлетворяет только евклидово расстояние. **(Справа)** Повернутый эллипс $\mathbf{x}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{S}^2\mathbf{R}\mathbf{x} = 1/4$, параллельный осям эллипс $\mathbf{x}^{\mathrm{T}}\mathbf{S}^2\mathbf{x} = 1/4$ и окружность $\mathbf{x}^{\mathrm{T}}\mathbf{x} = 1/4$ (**R** и **S**, как в примере 8.1)

если принять соглашение, что $x^0 = 0$ при $x = 0$ и 1 в остальных случаях. По сути дела, это расстояние с точки зрения ладьи: если клетка находится на другой горизонтали и вертикали, то до нее два хода, а если отличается только что-то одно, то один ход. Если **x** и **y** – строки нулей и единиц, то эта величина называется также *расстоянием Хэмминга*. Можно также сказать, что расстояние Хэмминга – это количество битов, которые нужно поменять, чтобы преобразовать **x** в **y**; для произвольных строк необязательно равной длины обобщение этой идеи приводит к *расстоянию Левенштейна*, или *редакционному расстоянию*.

Все ли описанные математические конструкции согласуются с нашим представлением о расстоянии? Чтобы ответить на этот вопрос, сформулируем, какими свойствами должно обладать настоящее расстояние, например: неотрицательность и симметричность. Общепризнанным является следующий список, определяющий так называемую метрику.

Определение 8.2 (метрика). *Если дано пространство объектов* X, *то* метрическим расстоянием, *или просто* метрикой, *в нем называется функция* Dis: $\mathbb{X} \times \mathbb{X} \to \mathbb{R}$ *такая, что для любых* $x, y, z \in \mathbb{X}$:

1. *Расстояние от точки до нее самой равно нулю:* $\mathrm{Dis}(x, x) = 0$;
2. *Все остальные расстояния больше нуля: если* $x \neq y$, *то* $\mathrm{Dis}(x, y) > 0$;
3. *Расстояние симметрично:* $\mathrm{Dis}(y, x) = \mathrm{Dis}(x, y)$;
4. *Объезд не может сократить расстояние:* $\mathrm{Dis}(x, z) \leq \mathrm{Dis}(x, y) + \mathrm{Dis}(y, z)$.

Если второе условие ослабить до нестрогого неравенства, то есть разрешить Dis(*x*, *y*) *быть равным нулю, даже когда* $x \neq y$, *то функция* Dis *называется* псевдометрикой.

Последнее условие называется *неравенством треугольника* (или субаддитивностью, поскольку оно касается соотношения между расстоянием и сложением). На рис. 8.4 это понятие исследуется для расстояний Минковского различных порядков. Неравенство треугольника говорит, что расстояние от начала координат до точки C не больше суммы расстояний от начала координат до точки A (Dis(O, A)) и от A до C (Dis(A, C)). Точка B находится на том же расстоянии от A, что и C, независимо от используемой метрики, таким образом, Dis(O,A) + Dis(A,C) равно расстоянию от начала координат до B. Значит, если нарисовать окружность с центром в начале координат, проходящую через B, то в силу неравенства треугольника точка C не должна оказаться вне этой окружности. На левом рисунке (евклидово расстояние) мы видим, что точка B – единственная, в которой пересекаются окружности с центрами в начале координат и в A, и, стало быть, во всех остальных точках неравенство треугольника строгое.

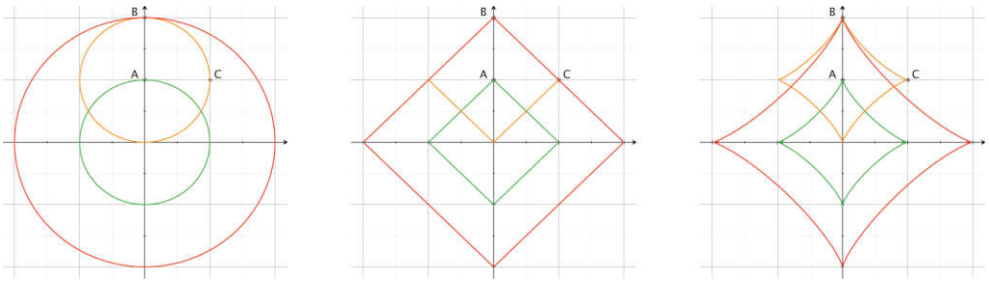

Рис. 8.4. (Слева) Зеленая окружность – геометрическое место точек, равноудаленных от начала координат в смысле евклидова расстояния (расстояния Минковского порядка $p = 2$), таких как A. **Оранжевая** окружность показывает, что B и C равноудалены от A. **Красная** окружность показывает, что C ближе к началу координат, чем B, что согласуется с неравенством треугольника. **(В центре)** В случае манхэттенского расстояния ($p = 1$) B и C расположены на равном расстоянии от начала координат и равноудалены также от A. **(Справа)** При $p < 1$ (в данном случае $p = 0.8$) C находится дальше от начала координат, чем B; поскольку обе точки по-прежнему равноудалены от A, то получается, что путешествие из начала координат в C через A быстрее, чем напрямую, что противоречит неравенству треугольника

На среднем рисунке та же ситуация показана для манхэттенского расстояния ($p = 1$). Теперь B и C равноудалены от начала координат, так что путешествие из A в C – уже не объезд, а лишь один из многих кратчайших путей. Но если мы будем еще уменьшать p, то в конце концов C окажется вне красной фигуры и, следовательно, дальше, чем B, если смотреть из начала координат; при этом сумма расстояний от начала координат до A и от A до C по-прежнему равна расстоянию от начала координат до B. В этот момент интуиция нам отказывает: расстояния

Минковского с $p < 1$ не слишком полезны в качестве расстояний, так как нарушают неравенство треугольника.

Иногда полезно применять разные шкалы для различных осей координат, если перемещение вдоль них происходит с разной скоростью. Например, люди перемещаются по горизонтали с куда большей легкостью, чем по вертикали, поэтому для определения точек, достижимых за определенное время, реалистичнее использовать не окружность, а эллипс, главная ось которого параллельна направлению наиболее быстрого перемещения. Этот эллипс можно также поворачивать, так что главная ось необязательно параллельна какой-то оси координат; например, это может быть направление шоссе или ветра. Математически гиперсфера (окружность в пространстве $d \geq 2$ измерений) радиуса r определяется уравнением $\mathbf{x}^T\mathbf{x} = r^2$, а гиперэллипс – уравнением $\mathbf{x}^T\mathbf{M}\mathbf{x} = r^2$, где $\mathbf{M}$ – матрица, описывающая поворот и масштабирование.

Пример 8.1 (эллиптическое расстояние). Рассмотрим следующие матрицы:

$$\mathbf{R} = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \quad \mathbf{S} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1 \end{pmatrix} \quad \mathbf{M} = \begin{pmatrix} 5/8 & -3/8 \\ -3/8 & 5/8 \end{pmatrix}.$$

Матрица $\mathbf{R}$ описывает поворот по часовой стрелке на 45°, а диагональная матрица $\mathbf{S}$ – масштабирование по оси x с коэффициентом 1/2. Уравнение

$$(\mathbf{SRx})^T(\mathbf{SRx}) = \mathbf{x}^T\mathbf{R}^T\mathbf{S}^T\mathbf{SRx} = \mathbf{x}^T\mathbf{R}^T\mathbf{S}^2\mathbf{Rx} = \mathbf{x}^T\mathbf{Mx} = 1/4$$

описывает фигуру, которая после поворота по часовой стрелке на 45° и масштабирования по оси x с коэффициентом 1/2 станет окружностью радиуса 1/2, то есть наклонный эллипс на рис. 8.3 справа. Уравнение этого эллипса: $(5/8)x^2 + (5/8)y^2 - (3/4)xy = 1/2$.

Часто форму эллипса оценивают по данным как обращение ковариационной матрицы: $\mathbf{M} = \boldsymbol{\Sigma}^{-1}$. Это приводит нас к определению *расстояния Махалонобиса*:

$$\mathrm{Dis}_M(\mathbf{x},\mathbf{y}|\boldsymbol{\Sigma}) = \sqrt{(\mathbf{x}-\mathbf{y})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\mathbf{y})}. \tag{8.1}$$

Как мы видели в разделе 7.1, такое применение ковариационной матрицы означает устранение корреляции признаков и их нормировку. Очевидно, что евклидово расстояние является частным случаем расстояния Махалонобиса, когда ковариационная матрица единичная: $\mathrm{Dis}_2(\mathbf{x},\mathbf{y}) = \mathrm{Dis}_M(\mathbf{x},\mathbf{y}|\mathbf{I})$.

8.2 Соседи и эталоны

Теперь, разобравшись с основами измерения расстояний в пространстве объектов, перейдем к рассмотрению ключевых идей, лежащих в основе метрических моделей. Наиболее важны две из них: формулировка модели в терминах типичных объектов, или *эталонов*, и определение решающего правила в терминах бли-

жайших эталонов, или *соседей*. Лучше понять эти концепции нам поможет наш старый приятель – базовый линейный классификатор. В нем эталонами являются два средних вектора классов $\mu^{\oplus}$ и $\mu^{\ominus}$, которые содержат все, что нам нужно знать об обучающих данных, чтобы построить классификатор. Фундаментальное свойство среднего вектора множества векторов заключается в том, что он доставляет минимум сумме квадратов евклидовых расстояний до всех векторов множества.

Теорема 8.1 (среднее арифметическое минимизирует сумму квадратов евклидовых расстояний). *Среднее арифметическое* μ *множества точек* D *в евклидовом пространстве является единственной точкой, в которой сумма квадратов евклидовых расстояний до этих точек достигает минимума.*

Доказательство. Мы покажем, что $\operatorname{argmin}_y \Sigma_{x\in D} \|\mathbf{x} - \mathbf{y}\|^2 = \mu$, где $\|\cdot\|$ обозначает 2-норму. Чтобы найти минимум, мы возьмем градиент (вектор частных производных по y_i) суммы и приравняем его к нулю:

$$\nabla_{\mathbf{y}} \sum_{\mathbf{x}\in D} \|\mathbf{x}-\mathbf{y}\|^2 = -2\sum_{\mathbf{x}\in D}(\mathbf{x}-\mathbf{y}) = -2\sum_{\mathbf{x}\in D}\mathbf{x} + 2|D|\mathbf{y} = 0,$$

откуда следует, что $\mathbf{y} = \frac{1}{|D|}\sum_{\mathbf{x}\in D}\mathbf{x} = \mu$.

Отметим, что минимизация суммы квадратов евклидовых расстояний до точек из заданного множества – то же самое, что минимизация *усредненной* суммы квадратов евклидовых расстояний. Может возникнуть вопрос, что случится, если опустить слово «квадратов»: ведь кажется более естественным взять в качестве эталона точку, минимизирующую просто сумму евклидовых расстояний. Такая точка называется *геометрической медианой*, потому что в случае одномерных данных она соответствует медиане, или «срединному значению» множества чисел. Однако в многомерном случае не существует замкнутой формулы для нахождения геометрической медианы, ее приходится вычислять методом последовательных приближений. Это вычислительное преимущество и есть основная причина, почему в метрических методах предпочитают использовать сумму квадратов евклидовых расстояний.

В некоторых ситуациях имеет смысл наложить на эталон ограничение: он должен совпадать с одной из заданных точек. В таком случае мы говорим о *медоиде*, чтобы отличить его от *центроида* – эталона, необязательно принадлежащего множеству данных. Для нахождения медоида мы должны для каждой точки вычислить сумму расстояний до всех остальных точек и выбрать ту точку, в которой эта сумма обращается в минимум. Безотносительно к используемой метрике в случае n точек для этого потребуется $O(n^2)$ операций, поэтому с точки зрения вычисления медоида нет причин предпочесть одну метрику другой. На рис. 8.5 показано множество из 10 точек, для которого различные способы определения эталона дают разные результаты. В частности, средняя точка и медоид, вычисленный с помощью квадрата 2-нормы, могут быть чрезмерно чувствительны к выбросам.

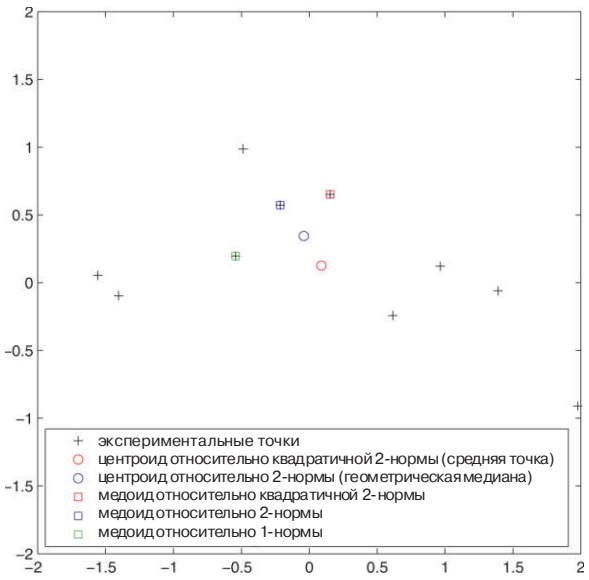

Рис. 8.5. Небольшой набор данных из 10 точек, кружочками обозначены центроиды, квадратиками – медоиды (которые должны совпадать с одной из имеющихся точек) для различных метрик. Обратите внимание, как выброс в правом нижнем углу «оттаскивает» среднюю точку от геометрической медианы, в результате изменяется и соответствующий медоид

Имея определение эталона, базовый линейный классификатор строит решающую границу в виде прямой, проходящей через середину соединяющего их отрезка перпендикулярно к нему. Альтернативный способ метрической классификации объектов без прямого упоминания решающей границы заключается в следующем правиле: если объект $\mathbf{x}$ ближе к $\mu^{\oplus}$, классифицировать его как положительный, иначе как отрицательный, или – эквивалентно – назначить объекту класс *ближайшего* к нему эталона. Если в качестве меры близости взять евклидово расстояние, то из простейших геометрических соображений следует, что мы получим в точности ту же самую решающую границу (рис. 8.6 слева).

Таким образом, *с метрической точки зрения, базовый линейный классификатор можно интерпретировать как построение эталонов, минимизирующих сумму квадратов расстояний в пределах каждого класса, и последующее применение решающего правила ближайшего эталона*. Такая смена угла зрения открывает массу новых возможностей. Например, можно исследовать, как будет выглядеть решающая граница, если в решающем правиле использовать манхэттенское расстояние (рис. 8.6 справа). Оказывается, что решающая граница может менять направление только под некоторыми углами: в двухмерном случае она может идти по горизонтали, по вертикали и под углом $\pm 45^{\circ}$. Это можно обосновать следующими рассуждениями. Предположим, что оба эталона имеют разные координаты x и y, то есть являются противоположными углами прямоугольника (я буду

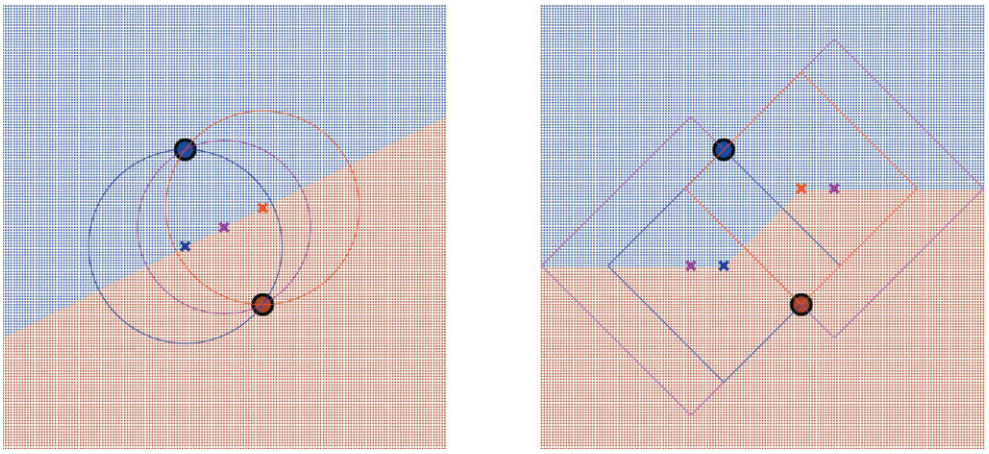

Рис. 8.6. (Слева) При двух эталонах решающее правило ближайшего эталона с евклидовым расстоянием дает линейную решающую границу – прямую, которая проходит через середину отрезка, соединяющего эталоны, перпендикулярно к нему. Крестиками отмечены точки на решающей границе, а окружности с центрами в этих точках наглядно демонстрируют, что они равноудалены от эталонов. При движении вдоль решающей границы из левого нижнего в правый верхний угол радиусы этих окружностей сначала уменьшаются, а после прохождения середины отрезка, соединяющего эталоны, снова начинают расти. **(Справа)** В случае использования манхэттенского расстояния окружности заменяются ромбами. При движении слева направо ромбы смещаются вдоль левого горизонтального участка решающей границы, уменьшаясь в размерах, затем движутся вдоль участка решающей границы, наклоненного под углом 45°, сохраняя размер, а потом снова смещаются вдоль горизонтального участка – правого

считать, что прямоугольник вытянут по вертикали, как на рисунке). Представьте, что вы стоите в центре этого прямоугольника и, следовательно, на равных расстояниях от обоих эталонов (на самом деле эта точка принадлежит решающей границе относительно 2-нормы). Если вы теперь отступите на один шаг по горизонтали, то приблизитесь к одному эталону и отдалитесь от другого; чтобы компенсировать это, необходимо сделать также шаг по вертикали. Таким образом, находясь внутри прямоугольника, вы сохраняете равноудаленность от эталонов, двигаясь под углом 45°. Достигнув периметра прямоугольника, вы можете обеспечить дальнейшее выполнение условия равноудаленности, только двигаясь по горизонтали, то есть решающая граница дальше будет проходить вертикально.

Еще одно полезное следствие взгляда на проблему с метрической точки зрения – тот факт, что решающее правило ближайшего эталона работает и тогда, когда эталонов больше двух. Это дает нам многоклассовый вариант базового линейного классификатора[1]. На рис. 8.7 слева это показано для трех эталонов.

[1] В контексте информационного поиска его часто называют *классификатором Роккио*.

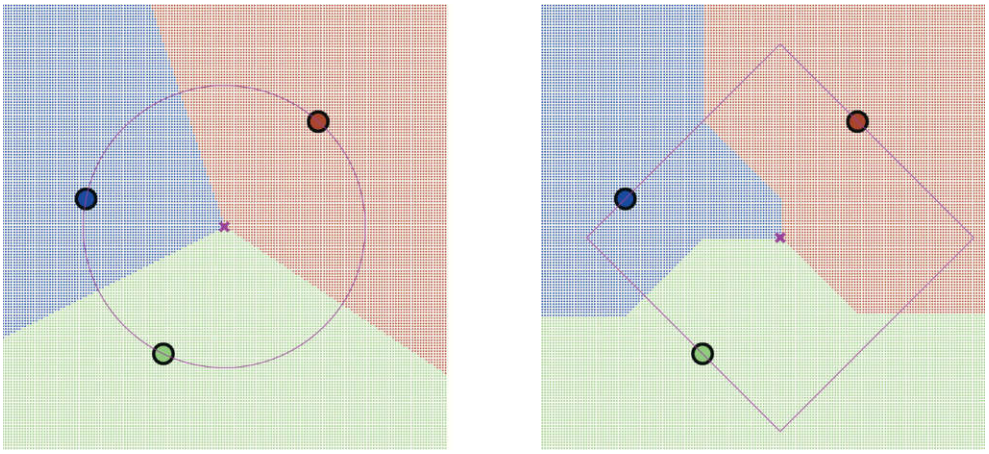

Рис. 8.7. (Слева) Решающие области, определенные решающим правилом ближайшего эталона относительно 2-нормы. **(Справа)** При использовании манхэттенского расстояния решающие области перестают быть выпуклыми

Каждая решающая область теперь ограничена двумя лучами. Как и следовало ожидать, решающие границы при использовании 2-нормы более регулярны, чем при использовании 1-нормы: математики говорят, что решающие области относительно 2-нормы выпуклы, то есть отрезок, соединяющий любые две точки, принадлежащие такой области, сам целиком принадлежит ей. Очевидно, что для решающих областей относительно 1-нормы это уже не так (рис. 8.7 справа). При дальнейшем увеличении числа эталонов некоторые области становятся замкнутыми выпуклыми «ячейками» (далее в этом разделе предполагается евклидово расстояние), что приводит к *диаграмме Вороного*. Поскольку число классов обычно гораздо меньше числа эталонов, решающие правила часто принимают во внимание больше одного эталона. В результате количество решающих областей еще увеличивается.

Пример 8.2 (два соседа знают больше, чем один). На рис. 8.8 слева показана диаграмма Вороного для пяти эталонов. Каждый отрезок лежит на прямой, которая проходит через середину отрезка, соединяющего два эталона, перпендикулярно к нему. Всего существует $\binom{5}{2} = 10$ пар эталонов, но в двух из них эталоны отстоят слишком далеко друг от друга, поэтому в диаграмме Вороного мы видим только восемь отрезков.

Если мы теперь примем во внимание еще и второй ближайший эталон, то каждая ячейка диаграммы разбивается на более мелкие. Например, поскольку у центральной точки четыре соседа, то содержащая ее ячейка разбивается на четыре подобласти (рис. 8.8 в центре). Эти дополнительные отрезки можно считать частями диаграммы Вороного, замещающими удаленную

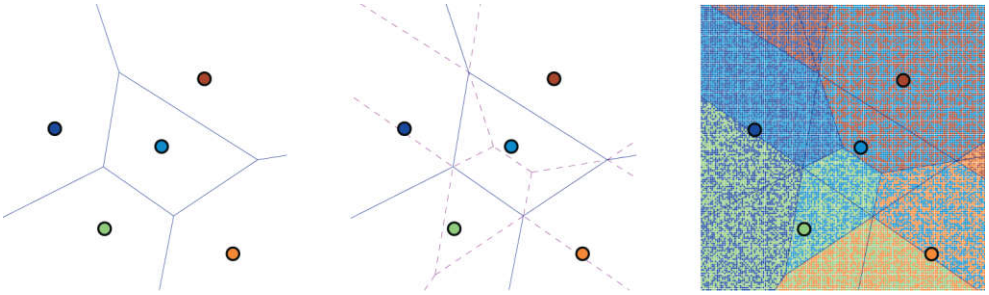

Рис. 8.8. (Слева) Диаграмма Вороного для пяти эталонов. **(В центре)** Если принимать во внимание два ближайших эталона, то каждая ячейка диаграммы Вороного разбивается на более мелкие. **(Справа)** Штриховкой показано, в какие ячейки вносит вклад каждый эталон

центральную ячейку. У других эталонов по три непосредственных соседа, поэтому их ячейки разбиваются на три подобласти. Таким образом, мы получаем 16 решающих областей «с 2 ближайшими эталонами», каждая из которых определяется своей парой ближайшего и второго по близости экземпляра.

На рис. 8.8 справа каждая область заштрихована в соответствии с тем, какие два ближайших эталона ее порождают. Отметим, что мы назначили обоим эталонам одинаковый вес и что пары соседних областей (граничащих по отрезкам исходной диаграммы Вороного) заштрихованы одинаково, так что всего понадобилось восемь разных видов штриховки. Это окажется существенным впоследствии, когда мы перейдем к обсуждению уточнения классификаторов по ближайшим соседям.

Итак, основными ингредиентами метрических моделей являются:

- метрические расстояния, в том числе евклидово, манхэттенское, Минковского и Махаланобиса;
- эталоны: центроиды, определяющие центр масс в соответствии с выбранной метрикой, или медоиды, определяющие «самую центральную» точку;
- метрические решающие правила, в которых проводится голосование среди k ближайших эталонов.

В следующих разделах мы увидим, как эти ингредиенты различными способами комбинируются для получения алгоритмов обучения с учителем и без.

8.3 Классификация по ближайшему соседу

В предыдущем разделе мы видели, как обобщить базовый линейный классификатор на число классов, большее двух, обучив эталон в каждом классе и используя решающее правило ближайшего эталона для классификации новых данных. На самом деле большинство популярных метрических классификаторов еще проще: в них эталоном может служить любой обучающий пример. Следовательно, «обучение» такого классификатора сводится просто к запоминанию обучающих

данных. Такой примитивный до крайности классификатор называется *классификатором по ближайшему соседу*. Его решающие области составлены из ячеек диаграммы Вороного, а кусочно-линейные решающие границы выбираются из множества границ диаграммы Вороного (поскольку соседние ячейки могут быть помечены одним и тем же классом).

Каковы свойства классификатора по ближайшему соседу? Прежде всего отметим, что если обучающий набор не содержит одинаковых объектов из разных классов, то мы сможем идеально разделить классы на обучающем наборе – и неудивительно, коль скоро мы запомнили все обучающие примеры! Далее за счет подходящего выбора эталонов мы можем представить более или менее любую решающую границу или, по крайней мере, сколь угодно точное кусочно-линейное приближение к ней. Отсюда следует, что классификатор по ближайшему соседу обладает низким смещением, но высокой дисперсией: сдвиньте любой эталон, порождающий решающую границу, – и сама граница тоже изменится. Отсюда риск переобучения, если обучающие данные ограничены, зашумлены или не репрезентативны.

С алгоритмической точки зрения, обучение классификатора по ближайшему соседу производится очень быстро – за время $O(n)$, где n – число хранимых эталонов. Недостаток в том, что и классификация одного экземпляра тоже занимает время $O(n)$, поскольку этот экземпляр необходимо сравнить с каждым эталоном, чтобы найти ближайший. Время классификации можно сократить за счет увеличения времени обучения путем хранения эталонов в более сложной структуре данных, но такой подход плохо масштабируется на большое число признаков.

На самом деле пространства объектов высокой размерности вызывают проблемы и по другой причине: печально известного *проклятия размерности*. Многомерные пространства обычно оказываются сильно разреженными, то есть каждая точка далеко отстоит практически от любой другой, а потому попарные расстояния не несут полезной информации. Однако настигнет вас проклятие размерности или нет, зависит не только от количества признаков, поскольку имеется ряд причин, из-за которых эффективная размерность пространства объектов может оказаться гораздо меньше числа признаков. Например, некоторые признаки могут быть нерелевантными и заглушать релевантные признаки при вычислении расстояний. В таких случаях полезно до построения метрической модели понизить размерность путем ☞ *отбора признаков*, который мы будем обсуждать в главе 10. Возможно также, что данные располагаются на *многообразии* более низкой размерности, чем размерность пространства объектов (например, на поверхности сферы, которая представляет собой двумерное многообразие, погруженное в трехмерное пространство), что позволяет применить другие методы понижения размерности, например ☞ *метод главных компонент*, рассматриваемый в той же главе. В любом случае, перед тем как применять классификацию по ближайшему соседу, имеет смысл построить гистограмму попарных расстояний между примерами в выборке и посмотреть, достаточно ли они разнообразны.

Отметим, что метод ближайшего соседа применим и к задачам регрессии с вещественнозначной целевой переменной. На самом деле этот метод абсолютно

безразличен к типу целевой переменной и может использоваться для классификации текстовых документов, изображений и видео. Можно также возвращать сам эталон, а не отдельную целевую переменную, и в таком случае обычно говорят о *поиске ближайшего соседа*. Разумеется, мы можем получать только цели (или эталоны), хранящиеся в базе данных эталонов, но если существует способ их агрегирования, то это ограничение можно снять, применив метод *k ближайших соседей*. В простейшей форме классификатор по k ближайшим соседям проводит голосование между $k \geq 1$ эталонами, ближайшими к классифицируемому объекту, и предсказывает класс, получивший большинство голосов (мажоритарный класс). Такой классификатор легко превратить в оценку вероятностей, возвращая нормированные счетчики классов в качестве распределения вероятностей по классам.

Рисунок 8.9 иллюстрирует это на небольшом наборе данных с 20 эталонами из пяти разных классов для $k = 3, 5, 7$. Распределение по классам наглядно представлено путем назначения каждой тестовой точке класса, определяемого соседями, не отдавая предпочтения ни одному из них. Так, в области, для которой два из $k = 3$ соседей красные, а один оранжевый, цвет штриховки на две трети состоит из красного и на одну треть из оранжевого. При $k = 3$ решающие области по большей части хорошо различимы, чего не скажешь о случаях $k = 5$ и $k = 7$. На первый взгляд, это противоречит ранее продемонстрированному в примере 8.2 увеличению числа решающих областей с ростом k. Однако это увеличение уравновешивается тем фактом, что векторы вероятностей становятся все больше похожи друг на друга. Возьмем крайний пример: если k равно числу эталонов n, то у каждого тестового объекта будет одинаковое число соседей, поэтому все они получат один и тот же вектор вероятностей, совпадающий с априорным распределением по эталонам. Если $k = n - 1$, то один из счетчиков классов можно уменьшить на 1, и сделать это можно c способами: число возможностей точно такое же, как при $k = 1$!

Мы делаем вывод, что уточнение метода k ближайших соседей – количество даваемых им различных предсказаний – сначала возрастает с ростом k, а затем

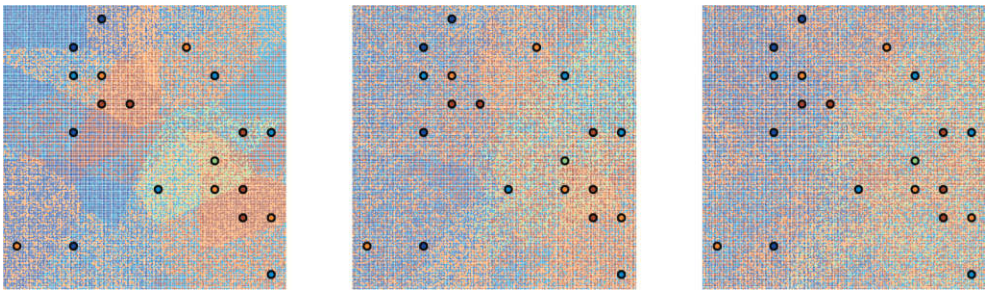

Рис. 8.9. (Слева) Решающие области для классификатора по 3 ближайшим соседям; штриховкой представлено предсказанное распределение вероятностей по пяти классам. **(В центре)** По 5 ближайшим соседям. **(Справа)** По 7 ближайшим соседям

снова убывает. Кроме того, мы можем сказать, что при увеличении k смещение растет, а дисперсия снижается. Не существует простого рецепта, который позволил бы решить, какое значение k лучше всего отвечает имеющемуся набору данных. Однако можно в какой-то степени обойти этот вопрос, применив к голосам *взвешивание расстояний*: то есть чем ближе эталон к классифицируемому объекту, тем весомее его голос. Это демонстрируется на рис. 8.10, где в качестве веса голоса используется величина, обратная расстоянию. При этом решающие границы размываются, поскольку теперь в модели используется сочетание группировки посредством границ диаграммы Вороного и ранжирования посредством взвешивания расстояний. Далее, поскольку веса быстро убывают с ростом расстояния, эффект увеличения k гораздо менее заметен, чем при голосовании без взвешивания. Фактически, если применяется взвешивание, мы можем просто положить $k = n$ и тем не менее получить модель, которая дает разные предсказания в разных частях пространства объектов. Можно сказать, что взвешивание расстояний делает классификатор по k ближайшим соседям в большей степени глобальной моделью, тогда как без него (и для малых k) это скорее агрегирование локальных моделей.

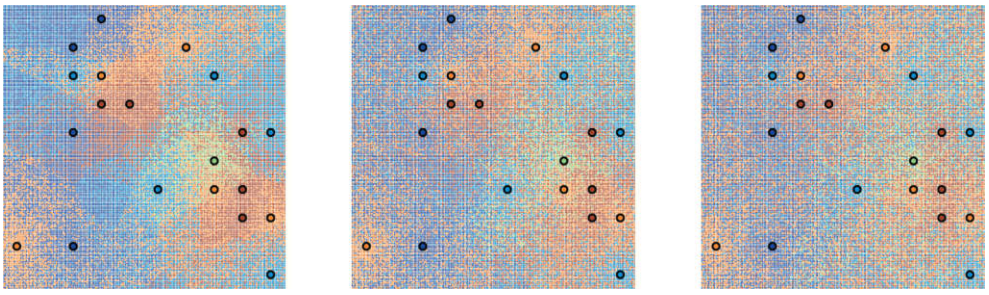

Рис. 8.10. (Слева) По 3 ближайшим соседям с взвешиванием расстояний на тех же данных, что на рис. 8.9. **(В центре)** По 5 ближайшим соседям. **(Справа)** По 7 ближайшим соседям

При использовании метода k ближайших соседей для задач регрессии очевидный способ агрегировать предсказания от k соседей заключается в том, чтобы взять среднее значение, которое опять-таки может быть взвешено с учетом расстояний. Это сообщило бы модели дополнительную предсказательную способность за счет предсказания значений, которые не наблюдались в хранимых эталонах. Более общо, мы можем применить метод k средних к любой проблеме обучения, в которой имеется подходящий «агрегатор» для нескольких целевых значений.

8.4 Метрическая кластеризация

В метрическом контексте обучение без учителя обычно связывают с кластеризацией, и сейчас мы дадим обзор нескольких метрических методов кластеризации.

Все методы, рассматриваемые в этом разделе, основаны на эталонах и потому являются прогностическими: они естественно обобщаются на ранее не предъявлявшиеся объекты (о различии между прогностической и дескриптивной кластеризациями см. раздел 3.3). В следующем разделе будет рассмотрен метод кластеризации, не основанный на эталонах и, следовательно, дескриптивный.

Прогностические метрические методы кластеризации состоят из тех же ингредиентов, что и метрические классификаторы: метрическое расстояние, способ построения эталонов и основанное на расстоянии решающее правило. В отсутствие явной целевой переменной предполагается, что цель обучения неявно закодирована в метрике, то есть мы стремимся найти кластеры, *компактные* относительно метрики. Для этого необходимо ввести понятие компактности кластера, которое может служить критерием оптимизации. С этой целью обратимся снова к матрице разброса, введенной в замечании 7.2 на стр. 211.

Определение 8.3 (разброс). *Если дана матрица данных* $\mathbf{X}$, *то* матрицей разброса *называется матрица*

$$\mathbf{S} = (\mathbf{X} - 1\mu)^{\top}(\mathbf{X} - 1\mu) = \sum_{i=1}^{n}(\mathbf{X}_i - \mu)^{\top}(\mathbf{X}_i - \mu),$$

где μ *– вектор-строка, содержащий средние по всем столбцам* $\mathbf{X}$. Разброс $\mathbf{X}$ *определяется как* $\mathrm{Scat}(\mathbf{X}) = \sum_{i=1}^{n}\|\mathbf{X}_i - \mu\|^2$, *то есть след матрицы разброса* (*сумма элементов на ее главной диагонали*).

Представим теперь, что мы разбили D на K подмножеств $D_1 \uplus \ldots \uplus D_K = D$, и пусть μ_j – среднее по D_j. Пусть $\mathbf{S}$ – матрица разброса D, а $\mathbf{S}_j$ – матрицы разброса D_j. Эти матрицы связаны следующим соотношением:

$$\mathbf{S} = \sum_{j=1}^{K}\mathbf{S}_j + \mathbf{B}. \tag{8.2}$$

Здесь $\mathbf{B}$ – матрица разброса, которая получается в результате замены каждой точки D соответствующим μ_j. Матрицы $\mathbf{S}_j$ называются *матрицами внутрикластерного разброса* и описывают компактность j-го кластера. $\mathbf{B}$ называется *матрицей межкластерного разброса* и характеризует разброс центроидов кластеров. Из (8.2) следует аналогичное разложение следов матриц:

$$\mathrm{Scat}(D) = \sum_{j=1}^{K}\mathrm{Scat}(D_j) + \sum_{j=1}^{K}|D_j|\|\mu_j - \mu\|^2. \tag{8.3}$$

Это говорит о том, что минимизация полного разброса по всем кластерам эквивалентна максимизации (взвешенного) разброса центроидов. *Проблема K средних* заключается в нахождении такого разбиения, которое минимизирует полный внутрикластерный разброс.

Пример 8.3 (уменьшение разброса путем разбиения данных). Рассмотрим такие пять точек: (0,3), (3,3), (3,0), (–2,–4), (–4,–2). Эти точки центрированы относительно (0,0) – очень удобно. Матрица разброса равна

$$\mathbf{S}=\begin{pmatrix}0&3&3&-2&-4\\3&3&0&-4&-2\end{pmatrix}\begin{pmatrix}0&3\\3&3\\3&0\\-2&-4\\-4&-2\end{pmatrix}=\begin{pmatrix}38&25\\25&38\end{pmatrix},$$

а ее след $\mathrm{Scat}(D) = 76$. Если поместить первые две точки в один кластер, а оставшиеся три – в другой, то средние векторы кластеров будут равны $\mu_1 = (1.5, 3)$ и $\mu_2 = (-1,-2)$, а матрицы внутрикластерного разброса:

$$\mathbf{S}_1=\begin{pmatrix}0-1.5&3-1.5\\3-3&3-3\end{pmatrix}\begin{pmatrix}0-1.5&3-3\\3-1.5&3-3\end{pmatrix}=\begin{pmatrix}4.5&0\\0&0\end{pmatrix};$$

$$\mathbf{S}_2=\begin{pmatrix}3-(-1)&-2-(-1)&-4-(-1)\\0-(-2)&-4-(-2)&-2-(-2)\end{pmatrix}\begin{pmatrix}3-(-1)&0-(-2)\\-2-(-1)&-4-(-2)\\-4-(-1)&-2-(-2)\end{pmatrix}=\begin{pmatrix}26&10\\10&8\end{pmatrix}$$

со следами $\mathrm{Scat}(D_1) = 4.5$ и $\mathrm{Scat}(D_2) = 34$. Две копии μ_1 и три копии μ_2 по определению имеют тот же центр тяжести, что полный набор данных: в данном случае (0,0). Таким образом, мы вычисляем матрицу межкластерного разброса как

$$\mathbf{B}=\begin{pmatrix}1.5&1.5&-1&-1&-1\\3&3&-2&-2&-2\end{pmatrix}\begin{pmatrix}1.5&3\\1.5&3\\-1&-2\\-1&-2\\-1&-2\end{pmatrix}=\begin{pmatrix}7.5&15\\15&30\end{pmatrix}$$

со следом 37.5.

С другой стороны, если рассматривать как кластер первые три точки, а остальные две поместить во второй кластер, то средние векторы кластеров будут равны $\mu'_1 = (2,2)$ и $\mu'_2 = (-3,-3)$, а матрицы внутрикластерного разброса:

$$\mathbf{S}'_1=\begin{pmatrix}0-2&3-2&3-2\\3-2&3-2&0-2\end{pmatrix}\begin{pmatrix}0-2&3-2\\3-2&3-2\\3-2&0-2\end{pmatrix}=\begin{pmatrix}6&-3\\-3&6\end{pmatrix};$$

$$\mathbf{S}'_2=\begin{pmatrix}-2-(-3)&-4-(-3)\\-4-(-3)&-2-(-3)\end{pmatrix}\begin{pmatrix}-2-(-3)&-4-(-3)\\-4-(-3)&-2-(-3)\end{pmatrix}=\begin{pmatrix}2&-2\\-2&2\end{pmatrix}$$

со следами $\mathrm{Scat}(D'_1) = 12$ и $\mathrm{Scat}(D'_2) = 4$. Матрица межкластерного разброса равна:

$$\mathbf{B}'=\begin{pmatrix}2&2&2&-3&-3\\2&2&2&-3&-3\end{pmatrix}\begin{pmatrix}2&2\\2&2\\2&2\\-3&-3\\-3&-3\end{pmatrix}=\begin{pmatrix}30&30\\30&30\end{pmatrix}$$

со следом 60. Очевидно, что второй способ кластеризации порождает более компактные кластеры, центроиды которых дальше отстоят друг от друга.

Алгоритм K средних

Проблема *K* средних является NP-полной, то есть эффективного способа найти глобальный минимум не существует, и приходится прибегать к эвристическим алгоритмам. Самый известный из них обычно называют точно так же: *K* средних, хотя встречается также название «алгоритм Ллойда». Его набросок приведен в алгоритме 8.1. Алгоритм поочередно разбивает данные, применяя решающее правило ближайшего центроида, и пересчитывает центроиды по разбиению. На рис. 8.11 этот алгоритм продемонстрирован на небольшом наборе данных с тремя кластерами, а в примере 8.4 приведен результат на наборе данных, описывающем свойства различных методов машинного обучения.

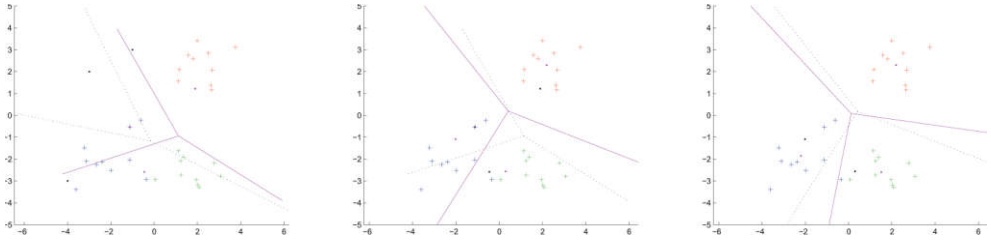

Рис. 8.11. (Слева) Первая итерация с 3 средними на данных с гауссовым распределением. Пунктирными линиями показаны границы диаграммы Вороного для центроидов, инициализированных случайным образом; **фиолетовые** сплошные линии – результат пересчета средних. **(В центре)** Вторая итерация, для которых начальными данными служит предыдущее разбиение. **(Справа)** Третья итерация с устойчивой кластеризацией

Пример 8.4 (кластеризация данных о методах машинного обучения). Обратимся к набору данных MMO о методах машинного обучения, показанному в табл. 1.4 на стр. 52 (полезно также взглянуть на его двухмерную аппроксимацию на рис. 1.7 на стр. 50). Прогнав алгоритм *K* средних на этих данных при $K = 3$, получим кластеры {Ассоциации, Деревья, Правила}, {GMM, наивная байесовская} и более крупный кластер, содержащий остальные точки. При $K = 4$ больший кластер разбивается на два: {kNN, линейный классификатор, линейная регрессия} и {K средних, логистическая регрессия, метод опорных векторов}; кроме того, GMM перемещается в последний кластер, а наивная байесовская классификация остается в одиночестве.

Можно показать, что одна итерация алгоритма *K* никогда не увеличивает внутрикластерного разброса, откуда следует, что алгоритм достигает *стационарного состояния*, после которого дальнейшее улучшение невозможно. Стоит отметить, что даже для простейшего набора данных стационарных состояний может быть много.

Алгоритм 8.1. KMeans(D, K) – кластеризация методом K средних с применением евклидова расстояния Dis_2

Вход: данные $D \subseteq \mathbb{R}^d$; число кластеров $K \in \mathbb{N}$.
Выход: средние K кластеров $\mu_1, ..., \mu_K \in \mathbb{R}^d$.

случайным образом инициализировать K векторов $\mu_1, ..., \mu_K \in \mathbb{R}^d$;
repeat
  отнести каждую точку $\mathbf{x} \in D$ к кластеру $\text{argmin}_j \text{Dis}_2(\mathbf{x}, \mu_j)$;
  for $j = 1$ до K **do**
    $D_j \leftarrow \{\mathbf{x} \in D | \mathbf{x}$ отнесена к кластеру $j\}$;
    $\mu_j = \frac{1}{|D_j|} \Sigma_{\mathbf{x} \in D_j} \mathbf{x}$;
  end
until $\mu_1, ..., \mu_K$ перестают изменяться;
return $\mu_1, ..., \mu_K$;

Пример 8.5 (стационарные состояния кластеризации). Рассмотрим задачу разбиения множества чисел {8,44,50,58,84} на два кластера. Всего возможны четыре разбиения, которые мог бы найти алгоритм 2 средних: {8}, {44,50,58,84}; {8,44}, {50,58,84}; {8,44,50}, {58,84} и {8,44,50,58}, {84}. Легко проверить, что каждое из них является стационарным состоянием алгоритма 2 средних и, следовательно, будет найдено при подходящей инициализации. Оптимальна только первая кластеризация, именно она минимизирует полный внутрикластерный разброс.

В общем случае, хотя алгоритм K средних сходится к стационарному состоянию за конечное время, не гарантируется, что найденное состояние является глобальным минимумом, и даже неизвестно, как далеко от него оно отстоит. На рис. 8.12 показано, что после неудачной инициализации центроидов может быть найдено неоптимальное решение. На практике рекомендуется прогнать алгоритм несколько раз и выбрать решение с наименьшим внутрикластерным разбросом.

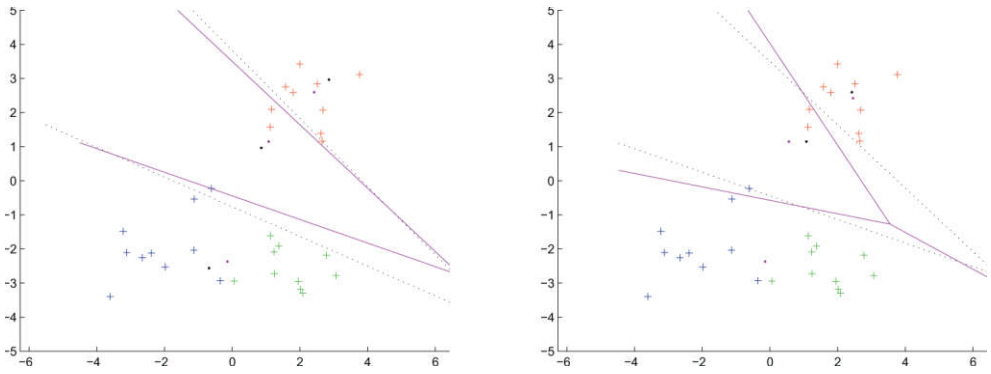

Рис. 8.12. (Слева) Первая итерация алгоритма 3 средних на тех же данных, что на рис. 8.11, но с центроидами, инициализированными иначе. **(Справа)** Алгоритм 3 средних сошелся к неоптимальной кластеризации

Кластеризация вокруг медоидов

Нетрудно модифицировать алгоритм K средних для другой метрики; отметим, что при этом изменится также минимизируемая целевая функция. В алгоритме 8.2 приведен алгоритм *K медоидов*, в котором дополнительно требуется, чтобы каждый эталон совпадал с какой-то точкой набора данных. Отметим, что для вычисления медоида кластера требуется перебрать все пары точек (тогда как для вычисления средней точки нужен лишь один проход по данным), для больших наборов это может оказаться вычислительно нереализуемым. Алгоритм 8.3 предлагает альтернативу – *разбиение по медоидам* (*PAM*), – в которой делается попытка локально улучшить кластеризацию, меняя медоиды местами с другими точками. Качество кластеризации Q вычисляется как сумма расстояний от каждой точки до ближайшего к ней медоида. Отметим, что существует $k(n-k)$ пар, состоящих из медоида и немедоида, а для вычисления Q необходимо перебрать $n-k$ точек, поэтому вычислительная сложность одной итерации квадратично зависит от количества точек в наборе данных. Для больших наборов можно прогнать алгоритм PAM на небольшой выборке, а Q посчитать по всему набору и повторить эту процедуру несколько раз для различных выборок.

Важным ограничением методов кластеризации, обсуждаемых в этом разделе, является тот факт, что они представляют кластеры только эталонами. При этом не учитывается форма кластеров, и иногда это приводит к результатам, противоречащим интуиции. Два набора данных на рис. 8.13 отличаются только масштабом по оси y. Тем не менее алгоритм K средних находит для них совершенно разные кластеризации. Это нельзя считать недостатком алгоритма K средних как такового, поскольку на рис. 8.13 справа оба центроида отстоят дальше друг от друга, чем в предполагаемом решении, и потому представляют лучшее решение в смысле критерия (8.3). Настоящая же проблема заключается в том, что в данном случае мы хотели бы оценить не только центроиды, но и «форму» кластеров, а значит, принимать во внимание не только следы матриц разброса. Мы обсудим эту тему в следующей главе.

Алгоритм 8.2. KMedoids(D, K, Dis) – кластеризация методом K медоидов с произвольной метрикой Dis

Вход: данные $D \subseteq \mathscr{X}$; число кластеров $K \in \mathbb{N}$; метрика $\text{Dis} : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$
Выход: K медоидов $\mu_1, ..., \mu_K \in D$, представляющие прогностическую кластеризацию $\mathscr{X}$.

случайным образом выбрать K точек $\mu_1, ..., \mu_K \in D$;
repeat
  отнести каждую точку $\mathbf{x} \in D$ к кластеру $\text{argmin}_j \text{Dis}(\mathbf{x}, \mu_j)$;
  for $j = 1$ до K **do**
    $D_j \leftarrow \{\mathbf{x} \in D | \mathbf{x}$ отнесена к кластеру $j\}$;
    $\mu_j = \text{argmin}_{x \in D_j} \sum_{\mathbf{x}' \in D_j} \text{Dis}(\mathbf{x}, \mathbf{x}')$;
  end
until $\mu_1, ..., \mu_K$ перестают изменяться;
return $\mu_1, ..., \mu_K$;

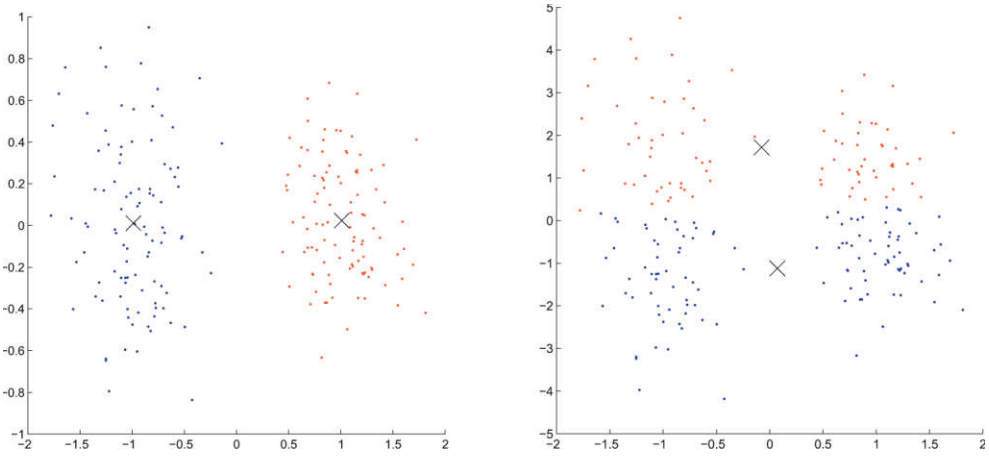

Рис. 8.13. (Слева) На этих данных алгоритм 2 средних обнаруживает правые кластеры. **(Справа)** После изменения масштаба по оси y у этой конфигурации межкластерный разброс оказался выше, чем у предполагаемой

Алгоритм 8.3. PAM(D, K, Dis) – кластеризация методом разбиения по медоидам с произвольной метрикой Dis

Вход: данные $D \subseteq \mathscr{X}$; число кластеров $K \in \mathbb{N}$; метрика $\text{Dis} : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$.
Выход: K медоидов $\mu_1, ..., \mu_K \in D$, представляющие прогностическую кластеризацию $\mathscr{X}$.

случайным образом выбрать K точек $\mu_1, ..., \mu_K \in D$;
repeat
  отнести каждую точку $\mathbf{x} \in D$ к кластеру $\text{argmin}_j \text{Dis}(\mathbf{x}, \mu_j)$;
  for $j = 1$ до K **do**
    $D_j \leftarrow \{\mathbf{x} \in D | \mathbf{x}$ отнесена к кластеру $j\}$;
  end
  $Q = \sum_j \sum_{\mathbf{x} \in D_j} \text{Dis}(\mathbf{x}, \mu_j)$;
  for каждого медоида **m** и каждого немедоида **o** **do**
    вычислить улучшение Q в результате замены **m** на **o**;
  end
  выбрать пару, дающую максимальное улучшение, и произвести перестановку;
until дальнейшие улучшения невозможны;
return $\mu_1, ..., \mu_K$;

Силуэты

Как можно было бы обнаружить плохое качество кластеризации на рис. 8.13 справа? Интересный ответ дают силуэты. Для любой точки $\mathbf{x}_i$ обозначим $d(\mathbf{x}_i, D_j)$ среднее расстояние от $\mathbf{x}_i$ до точек в кластере D_j, а $j(i)$ – индекс кластера, которо-

му принадлежит $\mathbf{x}_i$. Пусть далее $a(\mathbf{x}_i) = d(\mathbf{x}_i, D_{j(i)})$ – среднее расстояние от точки $\mathbf{x}_i$ до точек в ее собственном кластере $D_{j(i)}$, и пусть $b(\mathbf{x}_i) = \min_{k \neq j(i)} d(\mathbf{x}_i, D_k)$ – среднее расстояние до точек в соседнем с ней кластере. Мы хотели бы, чтобы $a(\mathbf{x}_i)$ было значительно меньше, чем $b(\mathbf{x}_i)$, но гарантировать это нельзя. Поэтому можно взять разность $b(\mathbf{x}_i) - a(\mathbf{x}_i)$ как меру того, насколько «хорошо кластеризована» точка $\mathbf{x}_i$, и разделить эту величину на $b(\mathbf{x}_i)$, чтобы получилось число, меньшее или равное 1.

Однако может случиться, что $a(\mathbf{x}_i) > b(\mathbf{x}_i)$, и в этом случае разность $b(\mathbf{x}_i) - a(\mathbf{x}_i)$ будет отрицательна. Это ситуация, когда в среднем члены соседнего кластера ближе к точке $\mathbf{x}_i$, чем члены ее собственного кластера. В этом случае для получения нормированного значения мы делим разность на $a(\mathbf{x}_i)$. Таким образом, мы приходим к следующему определению:

$$s(\mathbf{x}_i) = \frac{b(\mathbf{x}_i) - a(\mathbf{x}_i)}{\max(a(\mathbf{x}_i), b(\mathbf{x}_i))}. \tag{8.4}$$

Затем для получения *силуэта* необходимо рассортировать и нанести на график $s(\mathbf{x})$ для каждого объекта, сгруппировав данные по кластерам. На рис. 8.14 показаны примеры для двух кластеризаций, изображенных на рис. 8.13. В данном конкретном случае мы использовали для построения силуэта квадрат евклидова расстояния, но сам метод применим и к другим метрикам. Отчетливо видно, что первая кластеризация гораздо лучше второй. В дополнение к графическому представлению мы можем вычислить средние значения силуэта в каждом кластере и по всему набору данных.

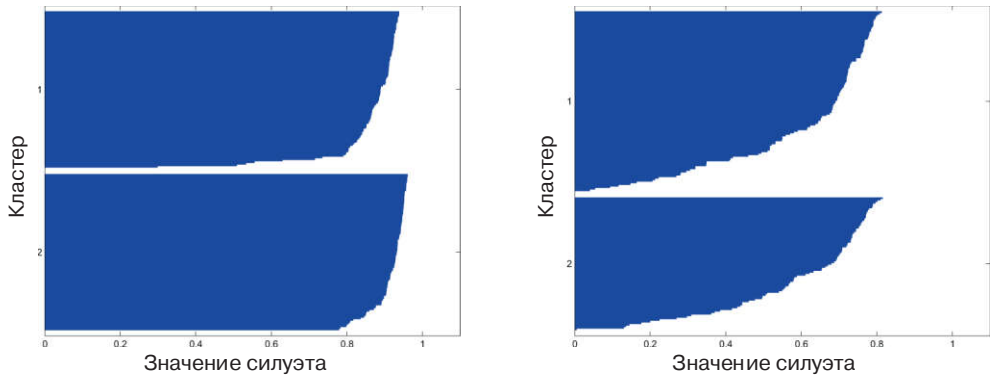

Рис. 8.14. (Слева) Силуэт для кластеризации, показанной на рис. 8.13 слева, с использованием евклидовой метрики. Почти для всех точек значение $s(\mathbf{x})$ велико, то есть в среднем они гораздо ближе к другим членам своего кластера, чем к членам соседнего кластера. **(Справа)** Силуэт для кластеризации, показанной на рис. 8.13 справа, гораздо менее убедителен

8.5 Иерархическая кластеризация

В методах, обсуждавшихся в предыдущем разделе, для представления прогностической кластеризации – разбиения всего пространства объектов – использовались эталоны. В этом разделе мы рассмотрим методы, позволяющие представить кластеры с использованием деревьев. Мы уже встречались с ☞ *кластеризующими деревьями* в разделе 5.3: в них для навигации по пространству объектов используются признаки, как в решающих деревьях, и от понятия расстояния они не зависят. А сейчас мы займемся деревьями, которые называются дендрограммами и определяются исключительно в терминах расстояния. Поскольку в дендрограммах признаки используются лишь косвенно, как основа для вычисления расстояний, то они разбивают предъявленные данные, а не все пространство объектов, и, следовательно, описывают не прогностическую, а дескриптивную кластеризацию.

Пример 8.6 (иерархическая кластеризация данных о методах машинного обучения). Продолжим пример 8.4. Иерархическая кластеризация набора MMO приведена на рис. 8.15. Дерево показывает, что все три логических метода сверху образуют сильно связанный кластер. Если бы мы запросили три кластера, то получили бы кластер логических методов, второй кластер поменьше {GMM, наивная байесовская} и все остальное. Если бы мы запросили четыре кластера, то GMM и наивная байесовская классификация разделились бы, поскольку, если верить дереву, этот кластер связан слабее прочих (отметим, что этот вывод несколько отличается от решения, найденного методом 4 средних). Если бы мы запросили пять кластеров, то был бы построен отдельный кластер {Линейная регрессия, Линейный классификатор}. Это иллюстрирует основное достоинство иерархической кластеризации: она не требует фиксировать количество кластеров заранее.

Точное определение дендрограммы формулируется так.

Определение 8.4 (дендрограмма). *Пусть имеется набор данных D,* дендрограммой *называется двоичное дерево, в котором листьями являются элементы D. Внутренний узел дерева представляет подмножество элементов в листьях поддерева с корнем в этом узле. Уровень узла – это расстояние между двумя кластерами, представленными потомками этого узла. Для листьев уровень равен 0.*

Чтобы это определение заработало, нам нужен способ измерить близость двух кластеров. На первый взгляд, все кажется простым: нужно лишь вычислить расстояние между средними точками кластеров. Однако иногда это приводит к проблемам, обсуждаемым ниже в этом разделе. Кроме того, взятие средних точек в качестве эталонов предполагает использование евклидовой метрики, а мы хотим, чтобы определение были пригодно и для других метрик. Все это привело к понятию функции связи – общего способа превратить попарные расстояния между точками в попарные расстояния между кластерами.

Определение 8.5 (функция связи). Функция связи $L\colon 2^{\mathscr{X}}\times 2^{\mathscr{X}} \to \mathbb{R}$ *вычисляет расстояние между произвольными подмножествами пространства объектов, если задана метрика* Dis: $\mathscr{X}\times\mathscr{X}\to\mathbb{R}$.

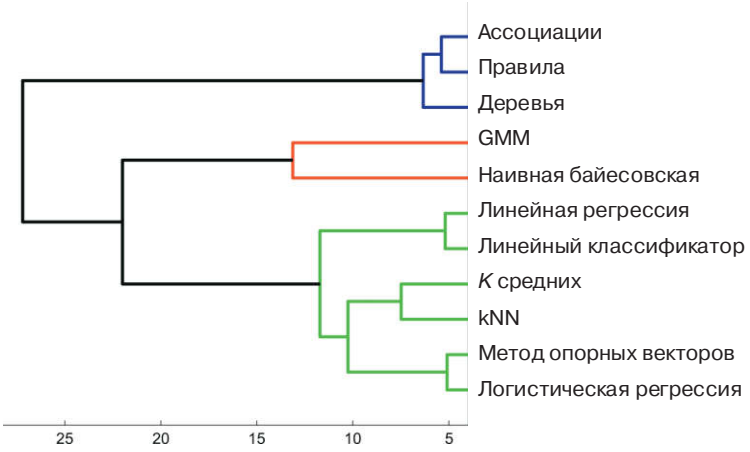

Рис. 8.15. Дендрограмма (напечатана слева направо для простоты восприятия), построенная по иерархической кластеризации данных в табл. 1.4 на стр. 52

Наиболее часто встречаются следующие функции связи.

Одиночная связь — определяет расстояние между двумя кластерами как *наименьшее* попарное расстояние между элементами, взятыми из разных кластеров.

Полная связь — определяет расстояние между двумя кластерами как *наибольшее* расстояние между точками из разных кластеров.

Средняя связь — определяет расстояние между двумя кластерами как *среднее* расстояние между точками из разных кластеров.

Центроидная связь — определяет расстояние между двумя кластерами как расстояние между средними точками кластеров.

Математически эти функции можно определить следующим образом:

$$L_{\text{single}}(A,B) = \min_{x\in A, y\in B} \text{Dis}(x,y);$$

$$L_{\text{complete}}(A,B) = \max_{x\in A, y\in B} \text{Dis}(x,y);$$

$$L_{\text{average}}(A,B) = \frac{\sum_{x\in A, y\in B} \text{Dis}(x,y)}{|A|\cdot|B|};$$

$$L_{\text{centroid}}(A,B) = \text{Dis}\left(\frac{\sum_{x\in A} x}{|A|}, \frac{\sum_{y\in B} y}{|B|}\right).$$

Понятно, что все эти функции связи дают один и тот же результат для кластеров, состоящих из одного элемента: $L(\{x\}, \{y\}) = \text{Dis}(x, y)$. Однако для более крупных кластеров они расходятся. Например, предположим, что $\text{Dis}(x, y) < \text{Dis}(x, z)$, тогда связь между $\{x\}$ и $\{y, z\}$ во всех четырех случаях различна:

$$L_{\text{single}}(\{x\}, \{y, z\}) = \text{Dis}(x, y);$$
$$L_{\text{complete}}(\{x\}, \{y, z\}) = \text{Dis}(x, z);$$
$$L_{\text{average}}(\{x\}, \{y, z\}) = (\text{Dis}(x, y) + \text{Dis}(x, z))/2;$$
$$L_{\text{centroid}}(\{x\}, \{y, z\}) = \text{Dis}(x, (y + z)/2).$$

Общий алгоритм построения дендрограммы описан в алгоритме 8.4. Дерево строится, начиная от точек набора данных, вверх, то есть это алгоритм восходящий, или, как еще говорят, *агломеративный*. На каждой итерации алгоритм строит новое разбиение данных, объединяя два ближайших кластера. В общем случае алгоритм HAC дает различные результаты при использовании разных функций связи. Проще всего понять его работу в случае одиночной связи, потому что при этом граф строится путем добавления все более длинных ребер между точками, по одному за раз, так что в конечном итоге будет существовать путь между любыми двумя точками (отсюда и термин «связь»). В любой момент этой процедуры связными компонентами графа являются найденные на текущей итерации кластеры, а связь последнего найденного кластера равна длине последнего добавленного ребра. Для выполнения иерархической кластеризации с применением одиночной связи нужно по существу вычислить и отсортировать все попарные расстояния между точками наборами данных, что требует времени порядка $O(n^2)$ для n точек. Для других функций связи требуется время не менее $O(n^2 \log n)$. Отметим, что сложность неоптимизированного алгоритма 8.4 составляет $O(n^3)$.

Алгоритм 8.4. HAC(D, L) – иерархическая агломеративная кластеризация

Вход: данные $D \subseteq \mathscr{X}$; функция связи $L: 2^{\mathscr{X}} \times 2^{\mathscr{X}} \to \mathbb{R}$, определенная в терминах метрики.
Выход: дендрограмма, представляющая дескриптивную кластеризацию D.

```
1 инициализировать кластеры, включив в каждый по одной точке;
2 создать узел уровня 0 для каждого такого кластера;
3 repeat
4 |   найти пару кластеров X, Y с наименьшей связью l и объединить их;
5 |   создать родителя X, Y на уровне l;
6 until все точки не окажутся в одном кластере;
7 return построенное двоичное дерево с уровнями связи;
```

Пример 8.7 (связь имеет значение). Рассмотрим регулярную сетку из 8 точек, расположенных в двух строках по четыре точки (рис. 8.16). Мы предполагаем, что неопределенности устранены за счет небольших нерегулярностей. Каждая функция связи объединяет одни и те же кластеры в одном и том же порядке, но сами связи в каждом случае различны. Полная связь создает впечатление, что D далеко отстоит от всех остальных, но если чуть-чуть подвинуть D вправо, то он был бы добавлен к E раньше C. В случае центроидной связи мы видим, что у E такая же связь, как у A и B, а это значит, что A и B не различимы как отдельные кластеры, хотя они и обнаруживаются первыми. В данном случае предпочтительной кажется одиночная связь, потому что она наиболее отчетливо демонстрирует, что никакой значимой кластерной структуры в этом наборе точек нет.

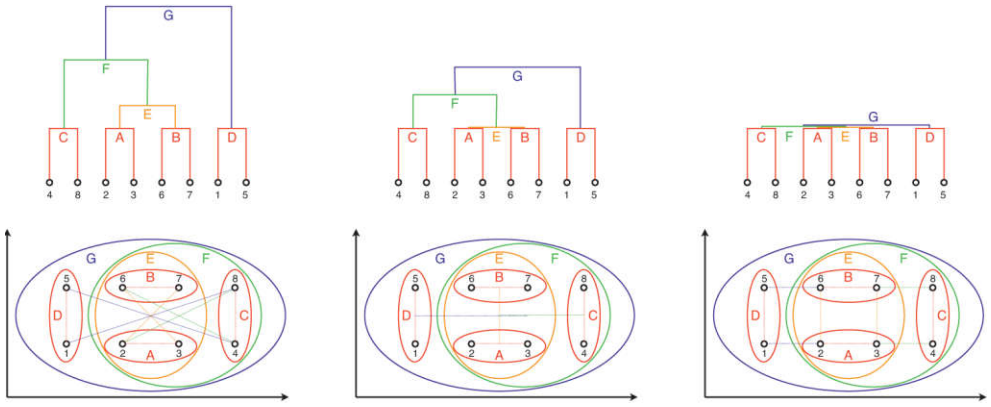

Рис. 8.16. (Слева) Полная связь определяет расстояние между кластерами как наибольшее попарное расстояние между их элементами; пары элементов показаны цветными линиями. Найденную кластеризацию можно представить в виде вложенных разбиений (снизу) или в виде дендрограммы (сверху); уровень горизонтального соединения между кластерами в дендрограмме соответствует длине связующего отрезка. В примере предполагается, что неопределенности устранены за счет небольших нерегулярностей сетки. **(В центре)** Центроидная связь определяет расстояние между кластерами как расстояние между их средними точками. Отметим, что у E оказывается такая же связь, как у A и B, поэтому два последних кластера по существу исчезают. **(Справа)** Одиночная связь определяет расстояние между кластерами как наименьшее попарное расстояние между их элементами. Дендрограмма практически схлопывается, то есть при данной конфигурации сетки обнаружить значимых кластеров не удалось

Одиночная и полная связи определяют расстояние между кластерами в терминах расстояния между определенной парой точек. Следовательно, они не могут учесть форму кластера, и именно поэтому средняя и центроидная связи могут оказаться предпочтительнее. Однако центроидная связь ведет к интуитивно неочевидным дендрограммам, как показано на рис. 8.17. Проблема в том, что $L(\{1\}, \{2\}) < L(\{1\}, \{3\})$ и $L(\{1\}, \{2\}) < L(\{2\}, \{3\})$, но $L(\{1\}, \{2\}) > L(\{1,2\}, \{3\})$. Первые два неравенства означают, что точки 1 и 2 должны быть объединены в кластер первыми, но третье неравенство говорит, что уровень кластера {1,2,3} в дендрограмме оказывается ниже уровня {1,2}. Центроидная связь нарушает требование *монотонности*, согласно которому из $L(A,B) < L(A,C)$ и $L(A,B) < L(B,C)$ должно следовать, что $L(A,B) < L(A \cup B,C)$ для любых кластеров A, B и C. Остальные три функции связи монотонны (пример также иллюстрирует, почему средняя и центроидная связь – не одно и то же).

При построении дендрограмм следует также иметь в виду, что иерархический метод кластеризации детерминирован и всегда завершается успешным построением кластеров. Взгляните на рис. 8.18, где показан набор из 20 случайных точек с равномерным распределением. Непредвзятому наблюдателю трудно обнаружить в этих данных хоть какую-нибудь кластерную структуру, однако дендрограмма,

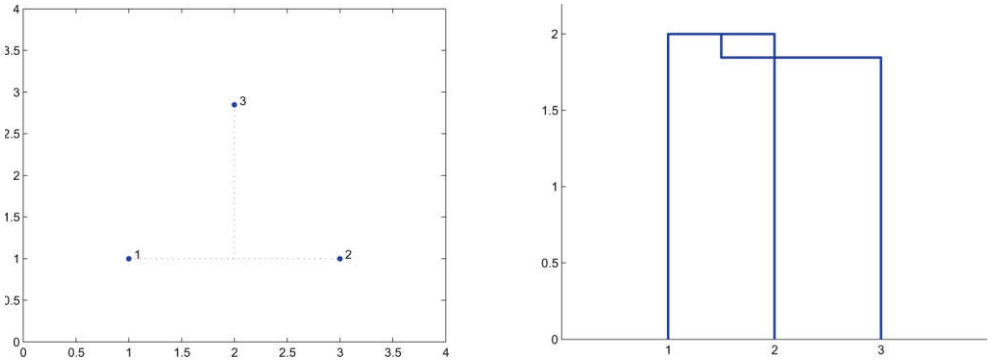

Рис. 8.17. (Слева) Точки 1 и 2 ближе друг к другу, чем к точке 3. Однако расстояние от точки 3 до центроида двух остальных точек меньше любого из попарных расстояний. **(Справа)** Это приводит к уменьшению связи после добавления точки 3 в кластер {1, 2}, а значит, к немонотонной дендрограмме

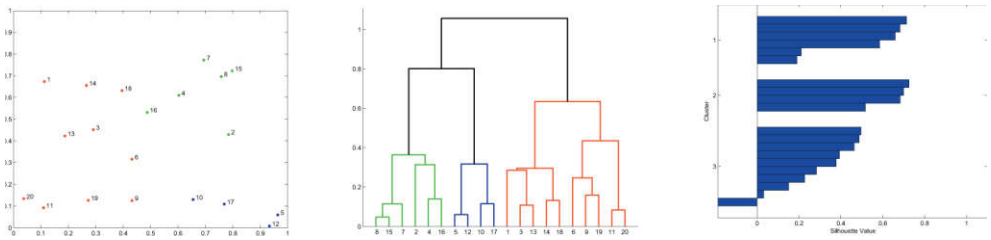

Рис. 8.18. (Слева) 20 случайных точек с равномерным распределением. **(В центре)** Дендрограмма, построенная с использованием полной связи. Все три предложенных ей кластера иллюзорные, поскольку в данных они не наблюдаются. **(Справа)** Быстрое убывание значений силуэта подтверждает отсутствие выраженной кластерной структуры. У точки 18 силуэт отрицательный, потому что в среднем она ближе к **зеленым** точкам, чем к остальным **красным**

построенная с помощью функции полной связи и евклидовой метрики, показывает существование трех или четырех отчетливых кластеров. Но, вглядевшись пристальнее, мы замечаем, что уровни связи очень близки друг к другу в нижней части дерева, а тот факт, что с ростом дерева они становятся выше, объясняется главным образом использованием полной связи, которая рассчитывается на основе максимального попарного расстояния. Силуэт на рис. 8.18 (справа) подтверждает, что кластерная структура не слишком развита. По существу, мы здесь наблюдаем своеобразную, свойственную кластеризации форму переобучения, уже знакомую нам по другим древовидным моделям, обсуждавшимся в главе 5. Кроме того, дендрограммы – как и другие древовидные модели – обладают высокой дисперсией, то есть небольшие изменения данных могут привести к значительному изменению дендрограммы.

В заключение отмечу, что иерархические методы кластеризации обладают тем несомненным достоинством, что количество кластеров заранее задавать не нужно. Однако за это приходится расплачиваться высокой стоимостью вычислений. И кроме того, теперь мы должны выбирать не только метрику, но и функцию связи.

8.6 От ядер к расстояниям

В разделе 7.5 мы обсудили, как с помощью ядер можно значительно расширить возможности линейных моделей. Напомним, что ядром называется функция $\boldsymbol{\kappa}(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$, которая вычисляет скалярное произведение в некотором пространстве признаков, не строя векторов признаков $\phi(\mathbf{x})$ явно. Любой метод обучения, который можно определить исключительно в терминах скалярных произведений обучающих примеров, пригоден для такого «перехода к ядру». В силу наличия тесной связи между евклидовой метрикой и скалярными произведениями этот «трюк с ядром» можно применить ко многим метрическим методам обучения.

В основе лежит тот факт, что евклидово расстояние можно записать в виде скалярных произведений:

$$\mathrm{Dis}_2(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_2 = \sqrt{(\mathbf{x}-\mathbf{y})\cdot(\mathbf{x}-\mathbf{y})} = \sqrt{\mathbf{x}\cdot\mathbf{x} - 2\mathbf{x}\cdot\mathbf{y} + \mathbf{y}\cdot\mathbf{y}}.$$

Из этой формулы с очевидностью следует, что расстояние между $\mathbf{x}$ и $\mathbf{y}$ убывает с ростом скалярного произведения $\mathbf{x}\cdot\mathbf{y}$, а это наводит на мысль, что и само скалярное произведение может служить мерой сходства. Однако оно не инвариантно относительно параллельных переносов, поскольку зависит от положения начала координат. В обеспечение такой инвариантности вносят вклад оба члена $\mathbf{x}\cdot\mathbf{x}$ и $\mathbf{y}\cdot\mathbf{y}$. Заменяя скалярное произведение ядром $\boldsymbol{\kappa}$, мы можем построить такое ядерное расстояние:

$$\mathrm{Dis}_{\kappa}(\mathbf{x},\mathbf{y}) = \sqrt{\kappa(\mathbf{x},\mathbf{x}) - 2\kappa(\mathbf{x},\mathbf{y}) + \kappa(\mathbf{y},\mathbf{y})}. \tag{8.5}$$

Как выясняется, Dis_{κ} определяет псевдометрику (см. определение 8.2), если $\boldsymbol{\kappa}$ – положительно полуопределенное ядро[1].

Для иллюстрации рассмотрим алгоритм 8.5, который адаптирует алгоритм *K* средних (алгоритм 8.1) к использованию ядерного расстояния. Таким образом, этот алгоритм производит кластеризацию согласно нелинейному расстоянию в пространстве объектов, которое соответствует евклидову расстоянию в неявном пространстве признаков. Однако возникает одна сложность: дело в том, что теорема 8.1 неприменима к нелинейным расстояниям, поэтому мы не можем постро-

[1] Она является метрикой, только если отображение признаков инъективно. Предположим, что это не так, тогда различные точки x и y отображаются в один и тот же вектор признаков $\phi(\mathbf{x}) = \phi(\mathbf{y})$, откуда следует, что $\boldsymbol{\kappa}(\mathbf{x},\mathbf{x}) - 2\boldsymbol{\kappa}(\mathbf{x},\mathbf{y}) + \boldsymbol{\kappa}(\mathbf{y},\mathbf{y}) = \phi(\mathbf{x})\cdot\phi(\mathbf{x}) - 2\phi(\mathbf{x})\cdot\phi(\mathbf{y}) + \phi(\mathbf{y})\cdot\phi(\mathbf{y}) = 0$.

ить средние точки кластеров в пространстве объектов. Поэтому в алгоритме 8.5 кластеризация трактуется как разбиение, а не как набор эталонов. Следовательно, отнесение каждой точки $\mathbf{x}$ к ближайшему кластеру (шаг 3) теперь является операцией квадратичной сложности, так как для каждого кластера необходимо вычислить сумму расстояний от $\mathbf{x}$ до всех его членов. Для алгоритма K средних сложность этого шага линейно зависит от $|D|$. Существует другой способ превратить скалярные произведения в расстояния. Поскольку скалярное произведение можно записать в виде $\|\mathbf{x}\| \cdot \|\mathbf{y}\| \cos\theta$, где θ – угол между векторами $\mathbf{x}$ и $\mathbf{y}$, то можно определить *косинусоидальную меру сходства* как

$$\cos\theta = \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\cdot\|\mathbf{y}\|} = \frac{\mathbf{x}\cdot\mathbf{y}}{\sqrt{(\mathbf{x}\cdot\mathbf{x})(\mathbf{y}\cdot\mathbf{y})}}. \tag{8.6}$$

Алгоритм 8.5. Kernel-KMeans(D, K) – кластеризация методом K средних с использованием ядерного расстояния Dis_κ

Вход: данные $D \subseteq \mathscr{X}$; число кластеров $K \in \mathbb{N}$.
Выход: K-разбиение $D_1 \uplus \ldots \uplus D_K = D$.

1 случайным образом инициализировать K кластеров $D_1, \ldots, D_K$;
2 **repeat**
3 отнести каждую точку $\mathbf{x} \in D$ к кластеру $\operatorname{argmin}_j \frac{1}{|D_j|}\sum_{\mathbf{y}\in D_j} \mathrm{Dis}_\kappa(\mathbf{x},\mathbf{y})$;
4 **for** $j = 1$ до K **do**
5 $D_j \leftarrow \{\mathbf{x} \in D | \mathbf{x}$ отнесена к кластеру $j\}$;
6 **end**
7 **until** $D_1, \ldots, D_K$ перестают изменяться;
8 **return** $D_1, \ldots, D_K$;

Косинусоидальное сходство отличается от евклидова расстояния тем, что не зависит от длины векторов $\mathbf{x}$ и $\mathbf{y}$. С другой стороны, оно не инвариантно относительно параллельных переносов, но придает специальное значение началу координат. Одна из возможных интерпретаций заключается в том, что векторы проецируются на единичную сферу с центром в начале координат, и расстояние между ними измеряется по поверхности этой сферы. Косинусоидальное сходство обычно преобразуют в метрику, рассматривая величину $1 - \cos\theta$. Будучи определено исключительно в терминах скалярных произведений, оно легко включается в ядерные алгоритмы, играя роль евклидовой метрики.