In [1]:
```
import IPython.display
IPython.display.display_latex(IPython.display.Latex(filename="../macros.tex"))
```

Метрические классификаторы

Гипотеза компактности — в задачах классификации предположение о том, что схожие объекты гораздо чаще лежат в одном классе, чем в разных; или, другими словами, что классы образуют компактно локализованные подмножества в пространстве объектов. Это также означает, что граница между классами имеет достаточно простую форму.

Имеем:
(X, Y) тренирововчная выборка размера N, количество признаков M
зафиксируем: $\rho : \hat{X}^2 \to [0, \infty)$ - функцию расстояния (она и нужна для формализации "близости/схожести" объектов)

1. $\rho(x_1, x_2) \geq 0$ неотрицательность
2. $\rho(x_1, x_2) = 0 \Leftrightarrow x_1 = x_2$ тождественность
3. $\rho(x, y) = \rho(y, x)$ симметричность
4. $\rho(x_1, x_3) \leq \rho(x_1, x_2) + \rho(x_2, x_3)$ неравенство треугольника

например евклидово расстояние:

$$\rho(u, x_i) = \sqrt{\sum_{j=0}^{M} (u^j - x_i^j)^2}$$

определим

- Для каждого элемента u, мы сортируем объекты по расстоянию до u (от близкого до дальнего). $x_{(u)}^{(i)}$ - i - ый сосед в выборке.
- C - конечное число классов.

обобщенный метрический классификатор

C - множество классов

$\Gamma_c(u) = \sum_{i=1}^{N}[y(x_{(u)}^{(i)}) = c] * w(i,u), c \in C$

$w(i,u)$ - вес (степень важности) i-ого соседа u. Неотрицательное и не возврастает по i.

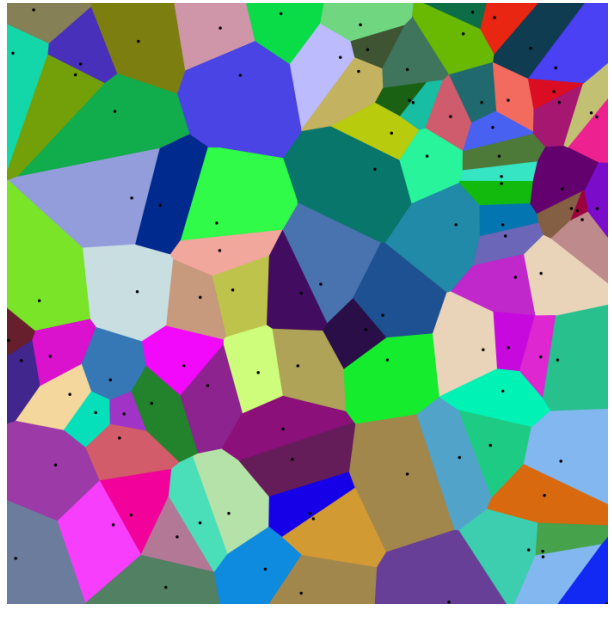
$$\alpha(u, X) = argmax_{c \in C} \sum_{i=1}^{N}[y(x_{(u)}^{(i)}) = c] * w(i,u)$$

Метод ближайшего соседа

$$w(i,u) = [i == 1]$$

- прост в реализации
- интерпретируемость решения
- неустойчивость к погрешностям/выбросам
- нет настраиваемых параметров
- низкое качество классификации
- необходимо хранить всю выборку целиком

Диаграмма (клетки) воронового



разделяющая поверхность - кусочно линейная функция

k-nearest neighbors algorithm

неустойчивость к погрешностям/выбросам

$$w(i,u) = [i < k+1]$$

Теперь есть возможность дать уровень уверенности принадледжать к классу.

- менее чувствителен к шуму
- появился параметр k

Проблема
$G_{c_1}(u) = G_{c_2}(u)$

Метод k взвешенных ближайших соседей

вес соседа не зависит от расстояния до него

$$w(i,u) = [i < k+1] * q(i)$$

q вес зависящий только от номера соседа.

Возможные эвристики::

- $q(i) = \frac{k+1-i}{k}$ - линейно убывающие веса
- $q(i) = (const)^i$ - экспоненциально убывающие веса $(0 < const < 1)$

Как более обоснованно задавать веса? Вес зависит только от порядкового номера соседа, возможно лучше было бы, если бы вес зависел и от расстояния до распознаваемого элемента

Метод окна Парзена

$$w(i,u) = K\Big(\frac{\rho(u, x_u^{(j)})}{h}\Big)$$

K - ядро

- невозрастающее
- положительно на $[0,1]$ равно нуле иначе:

$$|\rho(u, x_u^{(j)})| < |h|$$

h - ширина окна;

- Метод парзенского окна фиксированной ширины:

$$\alpha(u, X, h, K) = argmax_{c \in C} \sum_{i=1}^{N} [y_u^{(i)} = c] * K\left(\frac{\rho(u, x_u^{(j)})}{h}\right)$$

Плохо если обучающие объекты существенно неравномерно распределены по пространству. В окрестности одних объектов может оказываться очень много соседей, а в окрестности других — ни одного.

- Метод парзенского окна переменной ширины (финитное окно):

$$\alpha(u, X, k, K) = argmax_{c \in C} \sum_{i=1}^{N} [y_u^{(i)} = c] * K\left(\frac{\rho(u, x_u^{(j)})}{\rho\left(u, x_u^{(k+1)}\right)}\right)$$

выбор финитного ядра позволяет свести классификацию объекта u к поиску k его ближайших соседей, тогда как при не финитном ядре требуется полный перебор всей обучающей выборки

Примеры ядер

on $[0, 1]$

- равномерное: $\frac{1}{2}$
- треугольное: $(1 - \mid u \mid)$
- Епанечникова: $\frac{3}{4}(1 - u^2)$
- квадратическое: $\frac{15}{16}(1 - u^2)^2$
- Гауссовское: $\frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}u^2)$
- Косинусное: $\frac{\pi}{4} cos(\frac{\pi}{2}u)$

слабо влияет на качество классификации

Метод потенциальных функций

$$w(i,u) = \gamma^{(i)} K\Big(\frac{\rho(u, x_u^{(j)})}{h_i}\Big)$$

$$\alpha(u, X) = argmax_{c \in C} \sum_{i=1}^{N} [y_u^{(i)} = c] * \gamma^{(i)} * K\Big(\frac{\rho(u, x_u^{(j)})}{h_i}\Big)$$

$\gamma^{(i)}$ вес(потенциал) для $x_i \in X_{train}$

Физическая аналогия

- γ_i величина «заряда» в точке x_i
- h_i «радиус действия» потенциала с центром в точке

Алгоритм настройки весов объектов.

$K(r) = \frac{1}{r+a}$

FIT:

INPUT: X_{train}, Y size N
OUTPUT: γ_i for $i \in [1, N]$

START $\gamma_i = 0$ for $i \in [1, N]$
DO
chose $x_i \in X_{train}$
IF $\alpha(x_i) \neq y_i$ THEN $\gamma_i := \gamma_i + 1$
WHILE $\gamma(\alpha, X_{train}) < \epsilon$

Метод потенциальных функций

- Просто реализовать
- Не обязательно хранить всю выборку
- Медленно сходится
- Результат обучения зависит от порядка просмотра объектов
- Слишком грубо настраиваются веса
- Вообще не настраиваются hi
- Вообще не настраиваются центры потенциалов

Понятие отступа объекта

Отступом (margin) объекта $x_i \in X$ относительно алгоритма вида $\alpha(u, X) = argmax_{c \in C} \Gamma_c(u)$ называется величина $M(x_i) = \Gamma_{y_i}(x_i) - max_{c \in \{C \setminus y_i\}} \Gamma_c(x_i)$

Понятие «отступ» можно трактовать как «расстояние от объекта до поверхности, отделяющей свой класс от всех остальных».

$$M(x_i) < 0 \Leftrightarrow a(x_i) \neq y_i$$

- шумовые или выбросы - большой отрицательный отступ(окружён объектами чужих классов)
- эталонный представитель - большой положительный отступ означает(объект окружён объектами своего класса)
- пограничный - Отступ, близкий к нулю (классификация неустойчива в том смысле, что малые изменения в составе обучающей выборки могут приводить к ошибочной классификации объекта)
- неинформативные - В выборках избыточно большого объёма выделяется масса объектов с большим положительным отступом, которые правильно классифицируются по ближайшим к ним эталонам и фактически не несут никакой новой информации. Плотно окружены другими объектами того же класса.

Шумовые и неинформативные целесообразно удалять из выборки.

- повышается качество классификации
- сокращается объём хранимых данных
- уменьшается время классификации

Алгоритм STOLP

X^l — обучающая выборка;
δ — порог фильтрации выбросов;
l_0 — допустимая доля ошибок;

Вначале выкидываем шумовые по порогу фильтрации выбросов.

l_0 - допустимая доля ошибок.

Ω - множество опорных объектов
ИНИЦИИРУЕМ $\Omega = \{argmax_{x_i \in X_c^l} M(x_i, X^l) | c \in C\}$
ПОКА
$\Omega \neq X^l$ или $\left|\{x_i \in X^l \backslash \Omega : M(x_i, \Omega) < 0\}\right| < l_0$

Присоединяем к Ω объект с наименьшим отступом.