Метрические методы классификации

К. В. Воронцов, А.В. Зухба
vokov@forecsys.ru
a__l@mail.ru

февраль 2015

Содержание

1. Метрические алгоритмы классификации
   - Гипотеза компактности
   - Метод ближайших соседей и его обобщения
   - Метод парзеновского окна
   - Метод потенциальных функций
2. Отбор эталонов и оптимизация метрики
   - Понятие отступа
   - Алгоритм отбора эталонных объектов STOLP
3. Профиль компактности и скользящий контроль
   - Полный скользящий контроль CCV
   - Понятие профиля компактности
   - Отбор эталонов по функционалу CCV

Гипотеза компактности

Задача классификации:

X — объекты, Y — ответы (идентификаторы классов);
$X^\ell = (x_i, y_i)_{i=1}^\ell$ — обучающая выборка;

Гипотеза компактности:

Схожие объекты, как правило, лежат в одном классе.

Формализация понятия «сходства»:

Задана функция расстояния $\rho\colon X \times X \to [0, \infty)$.

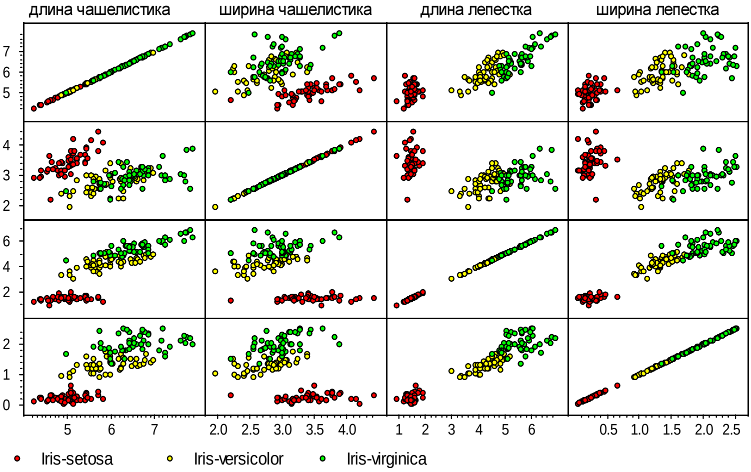
Например, евклидово расстояние:

$$\rho(u, x_i) = \left(\sum_{j=1}^{n} \left| u^j - x_i^j \right|^2 \right)^{1/2},$$

где $u = (u^1, \dots, u^n)$, $x_i = (x_i^1, \dots, x_i^n)$ — признаковые описания объектов.

Пример: задача классификации цветков ириса [Фишер, 1936]

$n = 4$ признака, $|Y| = 3$ класса, длина выборки $\ell = 150$.



Обобщённый метрический классификатор

Для произвольного $u \in X$ отсортируем объекты $x_1, \ldots, x_\ell$:

$$\rho(u, x_u^{(1)}) \leqslant \rho(u, x_u^{(2)}) \leqslant \cdots \leqslant \rho(u, x_u^{(\ell)}),$$

$x_u^{(i)}$ — i-й сосед объекта u среди $x_1, \ldots, x_\ell$;
$y_u^{(i)}$ — ответ на i-м соседе объекта u.

Метрический алгоритм классификации:

$$a(u; X^\ell) = \arg\max_{y \in Y} \underbrace{\sum_{i=1}^{\ell} \bigl[y_u^{(i)} = y\bigr]\, w(i, u)}_{\Gamma_y(u, X^\ell)},$$

$w(i, u)$ — вес (степень важности) i-го соседа объекта u, неотрицателен, не возрастает по i.

$\Gamma_y(u, X^\ell)$ — *оценка близости* объекта u к классу y.

Метод ближайшего соседа

$w(i, u) = [i{=}1]$.

Преимущества:

- простота реализации;
- интерпретируемость решений,
  вывод на основе прецедентов (case-based reasoning, CBR)

Недостатки:

- неустойчивость к погрешностям (шуму, выбросам);
- отсутствие настраиваемых параметров;
- низкое качество классификации;
- приходится хранить всю выборку целиком.

Метод k ближайших соседей

$w(i,u) = [i \leqslant k]$.

Преимущества:

- менее чувствителен к шуму;
- появился параметр k.

Оптимизация числа соседей k:

функционал скользящего контроля leave-one-out

$$\mathrm{LOO}(k, X^\ell) = \sum_{i=1}^{\ell} \Bigl[a\bigl(x_i; X^\ell \backslash \{x_i\}, k\bigr) \neq y_i \Bigr] \to \min_k .$$

Проблема:

- неоднозначность классификации при $\Gamma_y(u, X^\ell) = \Gamma_s(u, X^\ell)$, $y \neq s$.

Пример зависимости $LOO(k)$

Пример. Задача UCI: Breast Cancer (Wisconsin)

— смещённое число ошибок, когда объект учитывается как сосед самого себя
— несмещённое число ошибок LOO

В реальных задачах минимум редко бывает при $k = 1$.

Метод k взвешенных ближайших соседей

$w(i,u) = [i \leqslant k] w_i$,
где w_i — вес, зависящий только от номера соседа;

Возможные эвристики:

$w_i = \frac{k+1-i}{k}$ — линейное убывающие веса;
$w_i = q^i$ — экспоненциально убывающие веса, $0 < q < 1$;

Проблемы:

- как более обоснованно задать веса?
- возможно, было бы лучше, если бы вес $w(i,u)$ зависел не от порядкового номера соседа i, а от расстояния до него $\rho(u, x_u^{(i)})$.

Метод парзеновского окна

$w(i,u) = K\left(\frac{\rho(u,x_u^{(i)})}{h}\right)$,
где $K(r)$ — ядро, невозрастающее, положительное на $[0,1]$.

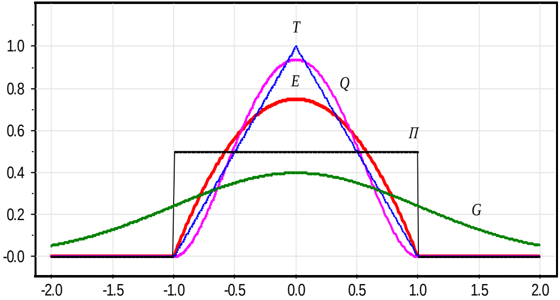
Метод парзеновского окна фиксированной ширины:

$$a(u;X^\ell,h,K) = \arg\max_{y\in Y} \sum_{i=1}^{\ell} [y_u^{(i)} = y]\, \underbrace{K\left(\frac{\rho(u,x_u^{(i)})}{h}\right)}_{w(i,u)}.$$

Метод парзеновского окна переменной ширины:

$$a(u;X^\ell,k,K) = \arg\max_{y\in Y} \sum_{i=1}^{\ell} [y_u^{(i)} = y]\, \underbrace{K\left(\frac{\rho(u,x_u^{(i)})}{\rho(u,x_u^{(k+1)})}\right)}_{w(i,u)}.$$

Часто используемые ядра



$E(r) = \frac{3}{4}(1 - r^2)\bigl[|r| \leqslant 1\bigr]$ — оптимальное (Епанечникова);
$Q(r) = \frac{15}{16}(1 - r^2)^2\bigl[|r| \leqslant 1\bigr]$ — квартическое;
$T(r) = \bigl(1 - |r|\bigr)\bigl[|r| \leqslant 1\bigr]$ — треугольное;
$G(r) = (2\pi)^{-1/2}\exp(-\frac{1}{2}r^2)$ — гауссовское;
$\Pi(r) = \frac{1}{2}\bigl[|r| \leqslant 1\bigr]$ — прямоугольное.

Метод потенциальных функций

$$w(i,u) = \gamma_u^{(i)} K\left(\frac{\rho(u,x_u^{(i)})}{h_u^{(i)}}\right)$$

Более простая запись:

$$a(u;X^\ell) = \arg\max_{y\in Y} \sum_{i=1}^{\ell} [y_i = y]\, \gamma_i\, K\left(\frac{\rho(u,x_i)}{h_i}\right),$$

где γ_i — веса объектов, $\gamma_i \geqslant 0$, $h_i > 0$.

Физическая аналогия:

γ_i — величина «заряда» в точке x_i;
h_i — «радиус действия» потенциала с центром в точке x_i;
y_i — знак «заряда» (предполагается, что $Y = \{-1,+1\}$);
в электростатике $K(r) = \frac{1}{r}$ или $\frac{1}{r+a}$.

Алгоритм настройки весов объектов

Простой эвристический алгоритм настройки γ_i.

Вход:

X^ℓ — обучающая выборка;

Выход:

Коэффициенты γ_i, $i = 1, \ldots, \ell$;

Инициализация: $\gamma_i = 0$ для всех $i = 1, \ldots, \ell$;
повторять
  выбрать объект $x_i \in X^\ell$;
  если $a(x_i) \neq y_i$ **то**
    $\gamma_i := \gamma_i + 1$;
пока число ошибок на выборке $Q(a, X^\ell) > \varepsilon$.

Анализ преимуществ и недостатков

Преимущества:

- простота реализации;
- не надо хранить выборку (потоковый алгоритм обучения);
- разреженность: не все обучающие объекты учитываются.

Недостатки:

- медленная сходимость;
- результат обучения зависит от порядка просмотра объектов;
- слишком грубо настраиваются веса γ_i;
- вообще не настраиваются параметры h_i;
- вообще не настраиваются центры потенциалов;
- может, некоторые γ_i можно было бы обнулить?

Понятие отступа

Рассмотрим классификатор $a\colon X \to Y$ вида

$$a(u) = \arg\max_{y \in Y} \Gamma_y(u), \quad u \in X.$$

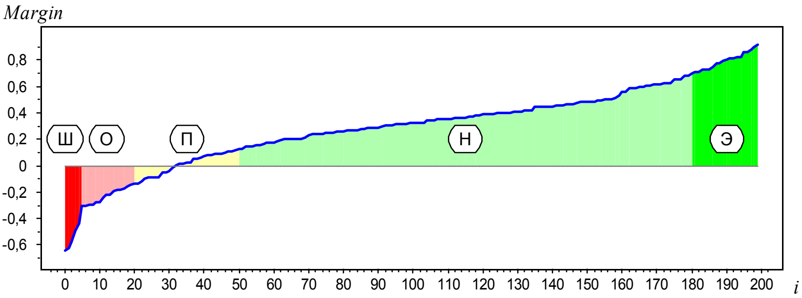
Отступом (margin) объекта $x_i \in X^\ell$ относительно классификатора $a(u)$ называется величина

$$M(x_i) = \Gamma_{y_i}(x_i) - \max_{y \in Y \setminus y_i} \Gamma_y(x_i).$$

- Отступ показывает *степень типичности* объекта: чем больше $M(x_i)$, тем «глубже» x_i в своём классе;
- $M(x_i) < 0 \;\Leftrightarrow\; a(x_i) \neq y_i$;

Типы объектов, в зависимости от отступа

- Э — *эталонные* (можно оставить только их);
- Н — *неинформативные* (можно удалить из выборки);
- П — *пограничные* (их классификация неустойчива);
- О — *ошибочные* (причина ошибки — плохая модель);
- Ш — *шумовые* (причина ошибки — плохие данные).



Отбор эталонов (prototype selection)

Задача: выбрать оптимальное подмножество эталонов $\Omega \subseteq X^\ell$

Классификатор будет иметь вид:

$$a(u;\Omega) = \arg\max_{y\in Y} \sum_{x_i\in\Omega} \bigl[y_u^{(i)} = y\bigr]\, w(i,u),$$

$x_u^{(i)}$ — i-й сосед объекта u среди Ω;
$y_u^{(i)}$ — ответ на i-м соседе объекта u;
$w(i,u)$ — произвольная функция веса i-го соседа.

Алгоритм STOLP:

1. исключить выбросы и, возможно, пограничные объекты;
2. найти по одному эталону в каждом классе;
3. добавлять эталоны, пока есть отрицательные отступы;

Алгоритм STOLP

Вход: X^ℓ; параметры δ, ℓ_0;
Выход: Множество опорных объектов $\Omega \subseteq X^\ell$;

1: **для всех** $x_i \in X^\ell$ проверить, является ли x_i выбросом:
2: **если** $M(x_i, X^\ell) < \delta$ **то**
3: $X^{\ell-1} := X^\ell \setminus \{x_i\}$; $\ell := \ell - 1$;
4: Инициализация: взять по одному эталону от каждого класса:
$\Omega := \Bigl\{\arg\max\limits_{x_i \in X^\ell_y} M(x_i, X^\ell) \Bigm| y \in Y\Bigr\}$;
5: **пока** $\Omega \neq X^\ell$;
6: Выделить множество объектов с ошибкой $a(u;\Omega)$:
$E := \{x_i \in X^\ell \setminus \Omega : M(x_i, \Omega) < 0\}$;
7: **если** $|E| < \ell_0$ **то выход**;
8: Присоединить к Ω объект с наименьшим отступом:
$x_i := \arg\min\limits_{x \in E} M(x, \Omega)$; $\Omega := \Omega \cup \{x_i\}$;

Алгоритм STOLP: преимущества и недостатки

Преимущества отбора эталонов:

- сокращается число хранимых объектов;
- сокращается время классификации;
- объекты распределяются по величине отступов;

Недостатки алгоритма STOLP:

- необходимость задавать параметр δ;
- относительно низкая эффективность $O(|\Omega|^2\ell)$.

Другие методы отбора:

- стратегия последовательного удаления не-эталонов;
- минимизация полного скользящего контроля (CCV);
- FRiS-STOLP на основе оценок *конкурентного сходства*.

Полный скользящий контроль CCV

Функционал *полного* скользящего контроля (complete cross-validation, CCV):

$$\mathrm{CCV}(X^L) = \frac{1}{C_L^\ell} \sum_{X^\ell \sqcup X^k} \frac{1}{k} \sum_{x_i \in X^k} \bigl[a(x_i, X^\ell) \neq y_i\bigr],$$

где $X^\ell \sqcup X^k$ — все C_L^ℓ разбиений выборки X^L на обучающую подвыборку X^ℓ и контрольную X^k.

Замечание 1. При $k = 1$ имеем: $\mathrm{CCV}(X^L) = \mathrm{LOO}(X^L)$.

Замечание 2. CCV характеризует лишь среднюю частоту ошибок, но не учитывает её разброс.

Понятие профиля компактности

Определение

Профиль компактности выборки X^L — это функция доли объектов x_i, у которых m-й сосед $x_i^{(m)}$ лежит в другом классе:

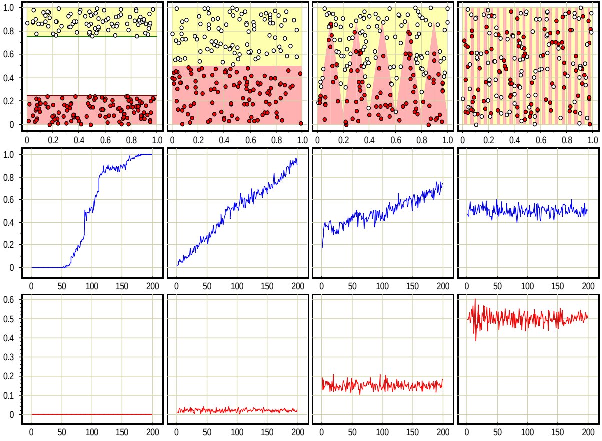
$$K(m, X^L) = \frac{1}{L} \sum_{i=1}^{L} \bigl[y_i \neq y_i^{(m)}\bigr]; \quad m = 1, \dots, L-1,$$

где $x_i^{(m)}$ — m-й сосед объекта x_i среди X^L;
$y_i^{(m)}$ — ответ на m-м соседе объекта x_i.

Теорема (точное выражение CCV для метода 1NN)

$$\mathrm{CCV}(X^L) = \sum_{m=1}^{k} K(m, X^L) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}}.$$

Профили компактности для серии модельных задач



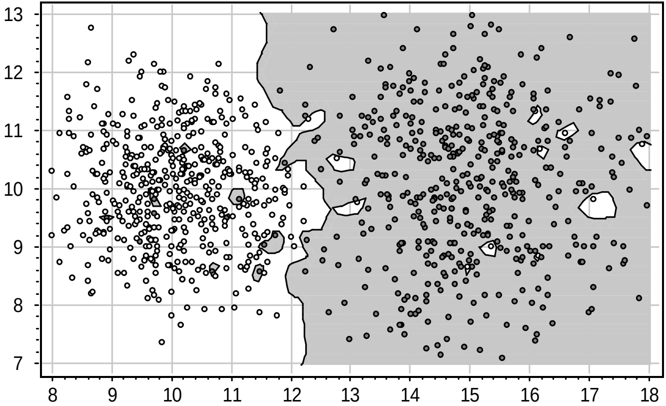
средний ряд: профили компактности,
нижний ряд: зависимость CCV от длины контроля k.

Свойства профиля компактности и оценки CCV

$$\mathrm{CCV}(X^L) = \sum_{m=1}^{k} K(m, X^L) \frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}}.$$

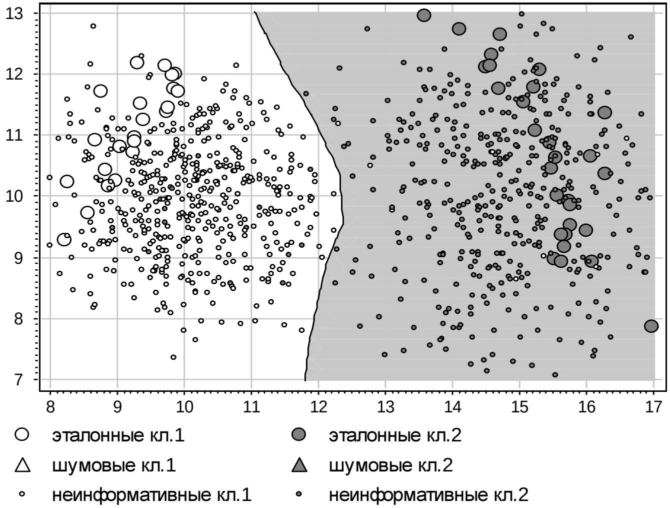
- $K(m, X^L)$ формализует гипотезу компактности, связывая свойства выборки с качеством классификации.
- CCV практически не зависит от длины контроля k.
- Для минимизации CCV важен только начальный участок профиля, т. к. $\frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} \to 0$ экспоненциально по m.
- Минимизация CCV позволяет делать отбор эталонов.

Модельные данные

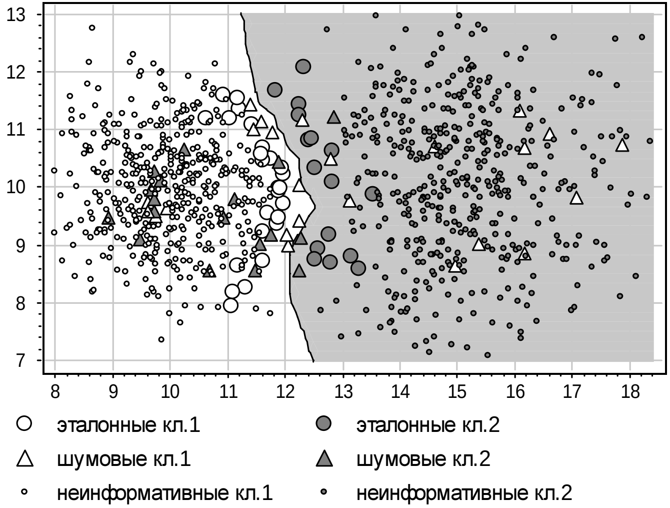


Модельная задача классификации: 1000 объектов.
Алгоритм 1NN

Последовательное добавление эталонных объектов

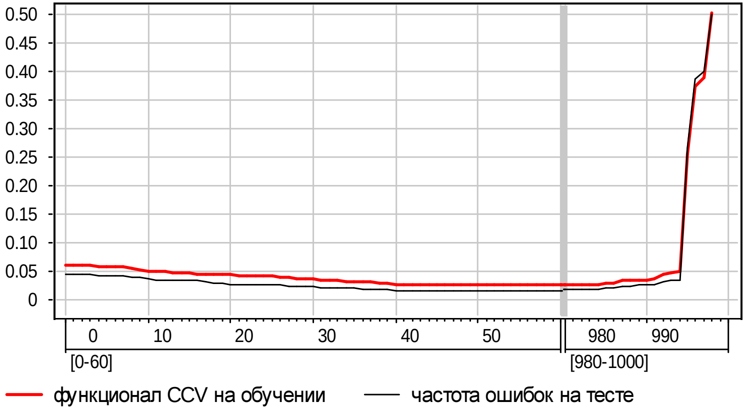


Последовательный отсев не-эталонных объектов



Последовательный отсев не-эталонных объектов

Зависимость CCV от числа удаленных неэталонных объектов.



При отборе эталонов по критерию CCV переобучения нет.

Резюме в конце лекции

- Метрические классификаторы — одни из самых простых. Качество классификации определяется качеством метрики.
- Что можно обучать:
  — число ближайших соседей k;
  — набор эталонов (prototype selection);
  — как вариант — веса объектов;
  — метрику (distance learning, similarity learning);
  — как частный случай — веса признаков.
- *Распределение отступов* делит объекты на эталонные, неинформативные, пограничные, ошибки и выбросы.
- *Профиль компактности* выборки позволяет судить о том, насколько удачно метрика подобрана под задачу.