

Метрические методы классификации

К. В. Воронцов
vokov@forecsys.ru

Этот курс доступен на странице вики-ресурса
<http://www.MachineLearning.ru/wiki>
«Машинное обучение (курс лекций, К.В.Воронцов)»

март 2011

Содержание

- 1 Метрические алгоритмы классификации**
 - Гипотеза компактности
 - Метод ближайших соседей и его обобщения
 - Снова метод парзеновского окна
 - Метод потенциальных функций
- 2 Отбор эталонов и оптимизация метрики**
 - Понятие отступа
 - Алгоритм отбора эталонных объектов STOLP
 - Понятие конкурентного сходства
 - Простой жадный алгоритм оптимизации метрики
- 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 $\rho: X \times X \rightarrow [0, \infty)$.

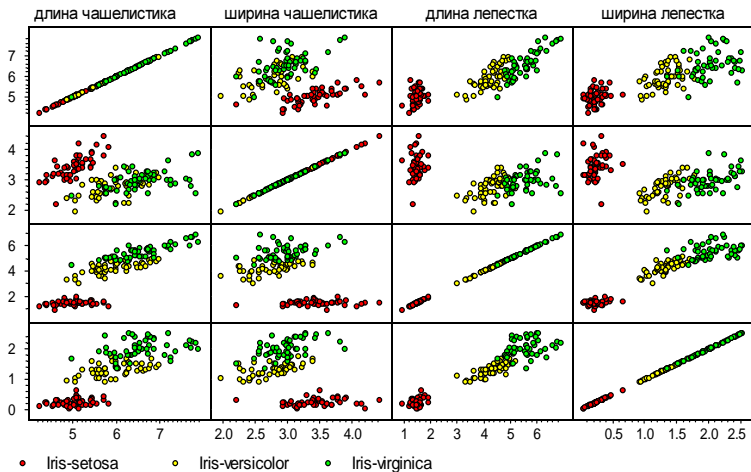
Например, евклидово расстояние:

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

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, \dots, x_ℓ :

$$\rho(u, x_u^{(1)}) \leq \rho(u, x_u^{(2)}) \leq \dots \leq \rho(u, x_u^{(\ell)}),$$

$x_u^{(i)}$ — i -й сосед объекта u среди x_1, \dots, x_ℓ ;

$y_u^{(i)}$ — ответ на i -м соседе объекта u .

Метрический алгоритм классификации:

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

$w(i, u)$ — 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

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

Преимущества:

- менее чувствителен к шуму;
- появился параметр k .

Оптимизация числа соседей k :

функционал скользящего контроля leave-one-out

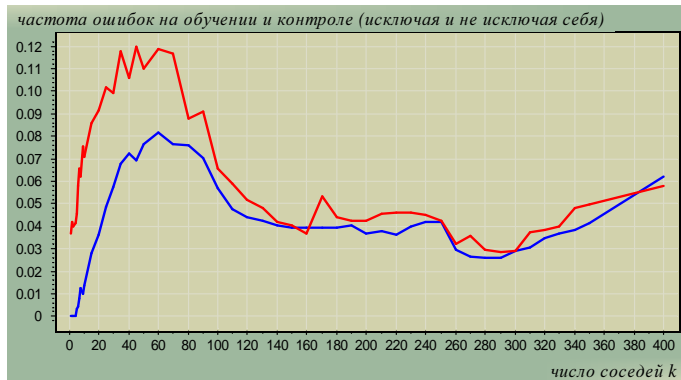
$$\text{LOO}(k, X^\ell) = \sum_{i=1}^{\ell} [a(x_i; X^\ell \setminus \{x_i\}, k) \neq y_i] \rightarrow \min_k.$$

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

$$w(i, u) = [i \leq k] w_i,$$

где w_i — вес, зависящий только от номера соседа;

Возможные эвристики:

$w_i = \frac{k+1-i}{k}$ — линейные 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 $\rho(u, x_u^{(i)})$.

Снова метод парзенковского окна

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

где $K(r)$ — ядро, невозрастающее, положительное на $[0, 1]$.

Метод парзенковского окна фиксированной ширины:

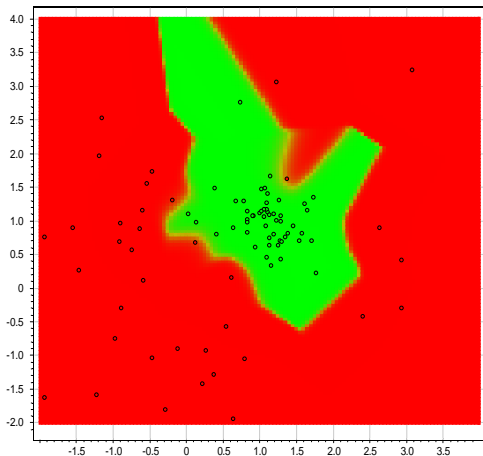
$$a(u; X^\ell, h, K) = \arg \max_{y \in Y} \sum_{i=1}^{\ell} [y_u^{(i)} = y] \underbrace{K\left(\frac{\rho(u, x_u^{(i)})}{h}\right)}_{w(i, u)}.$$

Метод парзенковского окна переменной ширины:

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

Метод парзеновского окна

Пример: классификация двумерной выборки.



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, $\gamma_i \geq 0$, $h_i > 0$.

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 , $i = 1, \dots, \ell$;

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

Анализ преимуществ и недостатков

Преимущества:

- простота реализации;
- не надо хранить выборку (поточковый 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?

Вывод: EM-RBF, конечно, круче...

Понятие отступа

Рассмотрим классификатор $a: X \rightarrow Y$ вида

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

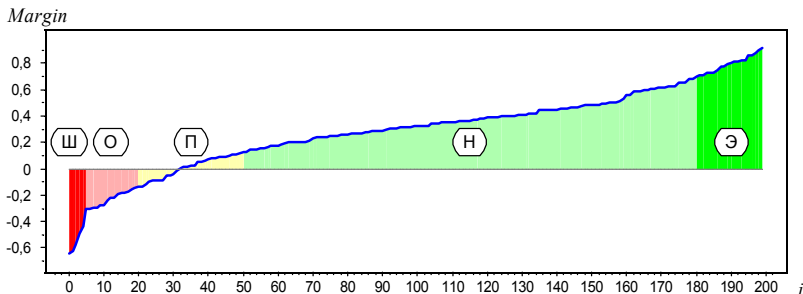
Отступом (margin) объекта $x_i \in X^\ell$ относительно классификатора $a(u)$ называется величина

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

- 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).



Типы объектов, в зависимости от отступа

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

Идея: шумовые и неинформативные удалить из выборки.

Алгоритм STOLP: основная идея

- исключить выбросы;
- найти по одному эталону в каждом классе;
- добавлять эталоны, пока есть отрицательные отступы;

Алгоритм STOLP

Вход:

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

Выход:

Множество опорных объектов $\Omega \subseteq X^\ell$;

Классификатор будет иметь вид:

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

- $x_u^{(i)}$ — 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: **для всех** $x_i \in X^\ell$ проверить, является ли x_i выбросом:
- 2: **если** $M(x_i, X^\ell) < \delta$ **то**
- 3: $X^{\ell-1} := X^\ell \setminus \{x_i\}; \quad \ell := \ell - 1;$
- 4: Инициализация: взять по одному эталону от каждого класса:
 $\Omega := \{\arg \max_{x_i \in X_y^\ell} M(x_i, X^\ell) \mid y \in Y\};$
- 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**
- 8: **выход;**
- 9: Присоединить к Ω объект с наименьшим отступом:
 $x_i := \arg \min_{x \in E} M(x, \Omega); \quad \Omega := \Omega \cup \{x_i\};$

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*.

Оценка близости i -го объекта к своему классу

Среднее расстояние до k ближайших объектов...

$r_i = r(x_i, y_i)$ — из своего класса;

$\bar{r}_i = r(x_i, \bar{y}_i)$ — из всех остальных классов;

Функция конкурентного сходства

(function of rival similarity, FRiS-функция)

$$d_i = \frac{\bar{r}_i - r_i}{\bar{r}_i + r_i} \approx \begin{cases} +1, & \text{объект близок к своим;} \\ 0, & \text{объект пограничный;} \\ -1, & \text{объект близок к чужим;} \end{cases}$$

Назовём d_i *благонадёжностью* объекта x_i .

Как и отступ, d_i — это характеристика типичности объекта относительно выборки.

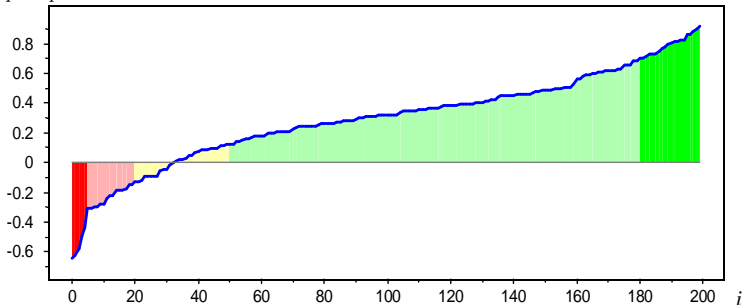
Преимущество — d_i величина безразмерная и нормированная.

Благонадёжность выборки

Суммарная благонадёжность выборки характеризует то, насколько функция расстояния ρ подходит для данной задачи

$$D(\rho) = \sum_{i=1}^{\ell} d_i = \sum_{i=1}^{\ell} \frac{\bar{r}_i - r_i}{\bar{r}_i + r_i}$$

распределение объектов по благонадёжности d_i



Жадное добавление признаков

1. А вдруг одного признака уже достаточно?

Расстояние по j -му признаку: $\rho_j(u, x_i) = |u^j - x_i^j|$.

Выберем наиболее благонадёжное расстояние: $D(\rho_j) \rightarrow \max_j$.

2. Пусть уже есть расстояние ρ .

Попробуем добавить к нему ещё один признак j .

$$\rho_{jt}(u, x_i) = (1 - t) \cdot \rho(u, x_i) + t \cdot \rho_j(u, x_i).$$

Найдём $t \in [0, 1]$ и признак j , при которых благонадёжность $D(\rho_{jt})$ максимальна (два вложенных цикла перебора).

3. Будем добавлять признаки до тех пор, пока благонадёжность $D(\rho_{jt})$ увеличивается.

Полный скользящий контроль CCV

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

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

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

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

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:

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

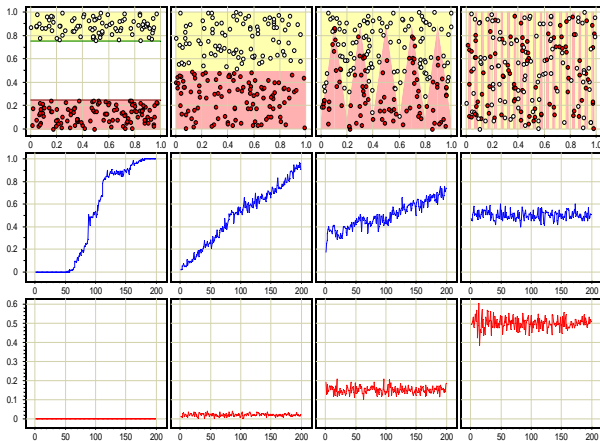
где $x_i^{(m)}$ — m -й сосед объекта x_i среди X^L ;

$y_i^{(m)}$ — ответ на m -м соседе объекта x_i .

Теорема (точное выражение CCV для метода 1NN)

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

Профили компактности для серии модельных задач



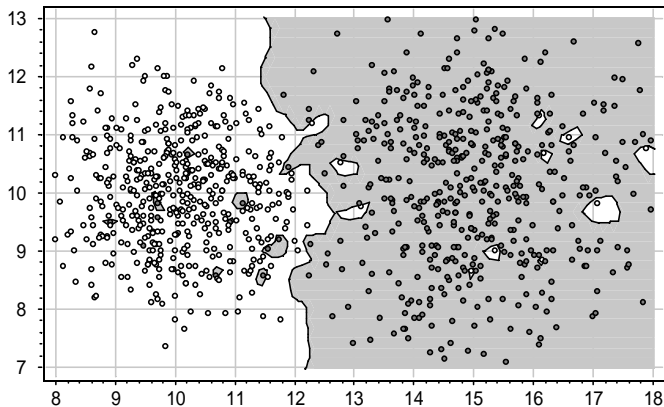
средний ряд: профили компактности,
нижний ряд: зависимость CCV от длины контроля q .

Свойства профиля компактности и оценки CCV

Выводы

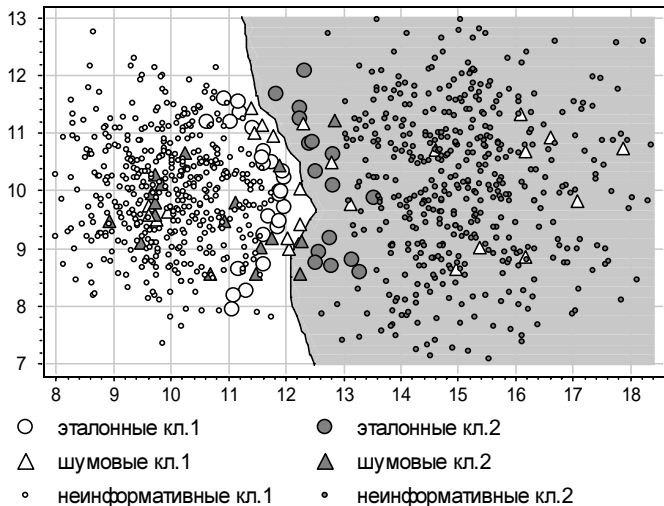
- $K(m, X^L)$ является формальным выражением гипотезы компактности, связывая её с качеством классификации.
- CCV практически не зависит от длины контроля q .
- Для минимизации CCV важен только начальный участок профиля, т. к. $\frac{C_{L-1-m}^{\ell-1}}{C_{L-1}^{\ell}} \rightarrow 0$ экспоненциально по m .
- Минимизация CCV приводит к эффективному отбору эталонных объектов, без переобучения.

Модельные данные

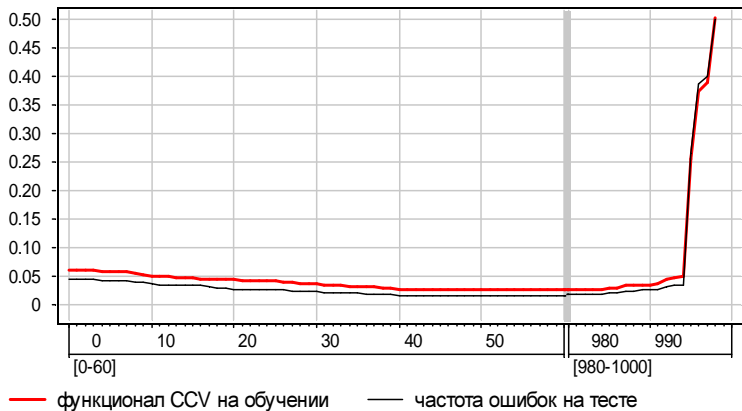


Модельная задача классификации: 1000 объектов.
Алгоритм 1NN

Последовательный отсев не-эталонных объектов



Последовательный отсев не-эталонных объектов



Зависимость CCV от числа удаленных неэталонных объектов.

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.