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)»

26 февраля 2009

Содержание

1. **Метрические алгоритмы классификации**
   - Гипотеза компактности
   - Методы типа ближайших соседей
   - Снова метод парзеновского окна
   - Метод потенциальных функций
2. **Отбор эталонных объектов**
   - Понятие отступа
   - Алгоритм отбора эталонных объектов

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:

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

Выборка задана $\ell \times \ell$ матрицей попарных расстояний (признаков вообще может не быть!)

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:

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

$w(i,u)$ — 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

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

Проблема:

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

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?
- как сделать, чтобы вес $w(i,u)$ зависел не от порядкового номера соседа i, а от расстояния до него $\rho(u, x_u^{(i)})$?

Снова 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:

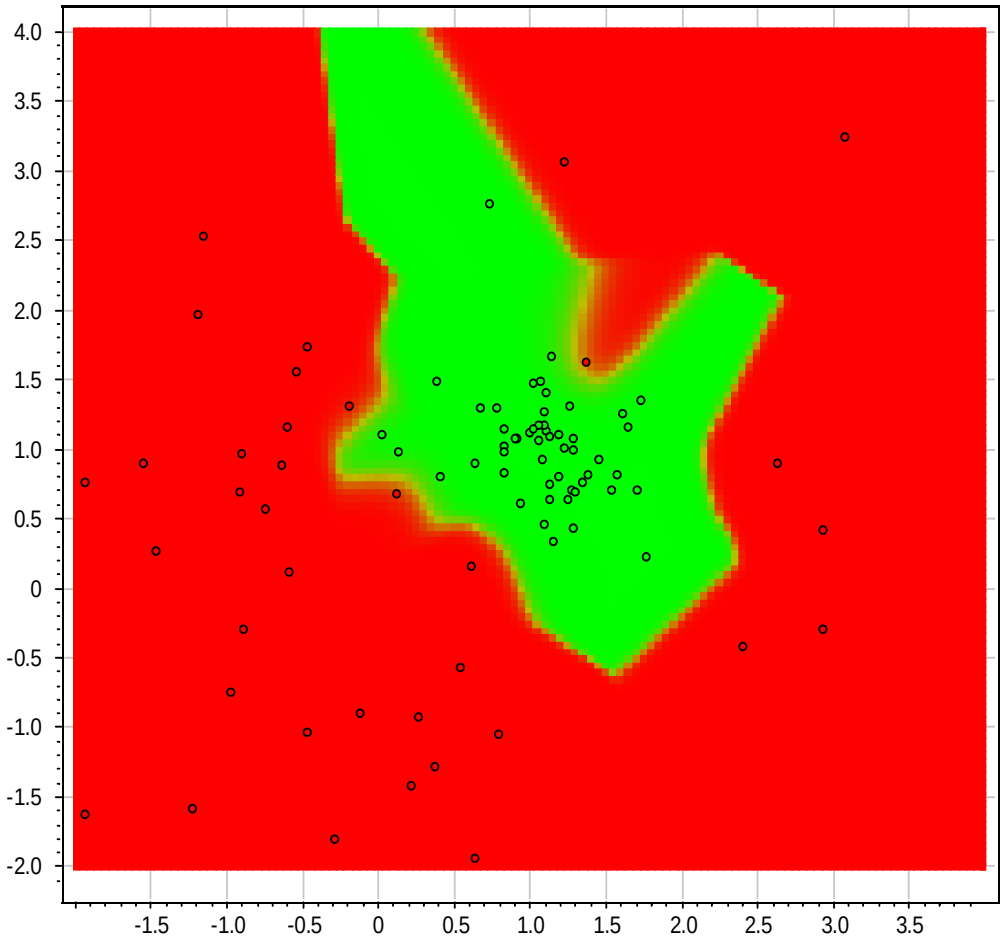
$$a(u; X^\ell, h, K) = \arg\max_{y\in Y} \sum_{i=1}^{\ell} [y_u^{(i)} = y] \underbrace{K\left(\frac{\rho(u,x_u^{(i)})}{h}\right)}_{w(i,u)}.$$

Метод парзеновского окна переменной ширины:

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

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

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

1: Инициализация: $\gamma_i = 0$ для всех $i = 1, \ldots, \ell$;
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 (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?

Вывод: EM-RBF, конечно, круче...

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

Типы объектов, в зависимости от отступа

- *эталонные* (можно оставить только их);
- *неинформативные* (можно удалить из выборки);
- *пограничные* (их классификация неустойчива);
- *ошибочные* (причина — плохая модель);
- *шумовые* (выбросы; причина — 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:

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

$x_u^{(i)}$ — i-й сосед объекта u среди Ω;

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

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:
$\Omega := \left\{ \arg\max_{x_i \in X^\ell_y} M(x_i, X^\ell) \;\middle|\; y \in Y \right\};$
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:

- универсальность (для любой функции $w(i,u)$);
- параметр δ управляет «шумоподавлением»;
- сокращается число хранимых объектов;
- сокращается время классификации;
- объекты распределяются по величине отступов;

Недостатки:

- относительно низкая эффективность $O(|\Omega|^2\ell)$.