Методы частичного обучения (semi-supervised learning)

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)»

28 октября 2015

Постановка задачи частичного обучения

Дано:

множество объектов X, множество классов Y;

$X^\ell = \{x_1, \ldots, x_\ell\}$ — размеченная выборка (labeled data);

$\{y_1, \ldots, y_\ell\}$

$X^k = \{x_{\ell+1}, \ldots, x_{\ell+k}\}$ — неразмеченная выборка (unlabeled data).

Два варианта постановки задачи:

- *Частичное обучение* (semi-supervised learning):
  построить алгоритм классификации $a\colon X \to Y$.
- *Трансдуктивное обучение* (transductive learning):
  зная все $\{x_{\ell+1}, \ldots, x_{\ell+k}\}$, получить метки $\{y_{\ell+1}, \ldots, y_{\ell+k}\}$.

Типичные приложения:

классификация и каталогизация текстов, изображений, и т. п.

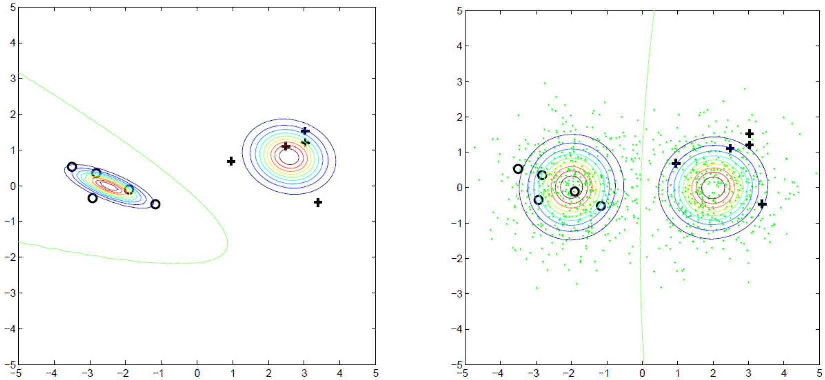
Содержание: методы кластеризации с частичным обучением

1. **Простые эвристические методы**
   - Особенности задачи SSL
   - Метод self-training
   - Композиции алгоритмов классификации
2. **Модификации методов кластеризации**
   - Оптимизационный подход
   - Кластеризация с ограничениями
   - Иерархическая кластеризация с ограничениями
3. **Модификации методов классификации**
   - Трансдуктивный SVM
   - Логистическая регрессия
   - Expectation Regularization

SSL не сводится к классификации

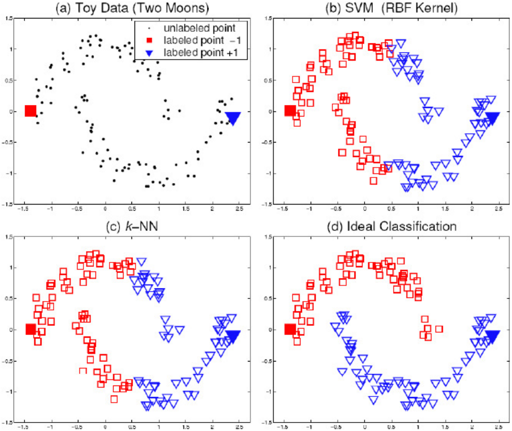
Пример 1. плотности классов, восстановленные:

по размеченным данным X^ℓ по полным данным $X^{\ell+k}$



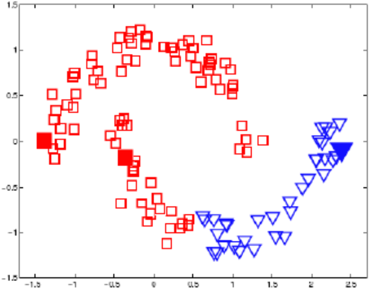
SSL не сводится к классификации

Пример 2. Методы классификации не учитывают кластерную структуру неразмеченных данных



Однако и к кластеризации SSL также не сводится

Пример 3. Методы кластеризации не учитывают приоритетность разметки.



Метод self-training (1965-1970)

Пусть $\mu\colon X^\ell \to a$ — произвольный метод обучения;
классификаторы имеют вид $a(x) = \arg\max\limits_{y\in Y} \Gamma_y(x)$;

Псевдоотступ — степень уверенности классификации $a_i = a(x_i)$:

$$M_i(a) = \Gamma_{a_i}(x_i) - \max_{y\in Y\setminus a_i} \Gamma_y(x_i).$$

Алгоритм self-training — обёртка (wrapper) над методом μ:

1: $Z := X^\ell$;
2: **пока** $|Z| < \ell + k$
3: $\quad a := \mu(Z)$;
4: $\quad \Delta := \bigl\{x_i \in X^k \setminus Z \bigm| M_i(a) \geqslant M_0\bigr\}$;
5: $\quad y_i := a(x_i)$ для всех $x_i \in \Delta$;
6: $\quad Z := Z \cup \Delta$;

M_0 можно определять, например, из условия $|\Delta| = 0.05\,k$

Метод co-training (Blum, Mitchell, 1998)

Пусть $\mu_1\colon X^\ell \to a_1$, $\mu_2\colon X^\ell \to a_2$ — два существенно различных метода обучения, использующих
— либо разные наборы признаков;
— либо разные парадигмы обучения (inductive bias);
— либо разные источники данных $X_1^{\ell_1}$, $X_2^{\ell_2}$.

$Z_1 := X_1^{\ell_1}$; $Z_2 := X_2^{\ell_2}$;
пока $|Z_1 \cup Z_2| < \ell + k$
  $a_1 := \mu_1(Z_1)$; $\Delta_1 := \bigl\{x_i \in X^k \backslash Z_1 \backslash Z_2 \bigm| M_i(a_1) \geqslant M_{01}\bigr\}$;
  $y_i := a(x_i)$ для всех $x_i \in \Delta_1$;
  $Z_2 := Z_2 \cup \Delta_1$;
  $a_2 := \mu_2(Z_2)$; $\Delta_2 := \bigl\{x_i \in X^k \backslash Z_1 \backslash Z_2 \bigm| M_i(a_2) \geqslant M_{02}\bigr\}$;
  $y_i := a(x_i)$ для всех $x_i \in \Delta_2$;
  $Z_1 := Z_1 \cup \Delta_2$;

Метод co-learning (deSa, 1993)

Пусть $\mu_t \colon X^\ell \to a_t$ — разные методы обучения, $t = 1, \ldots, T$.

Алгоритм co-learning — это self-training для композиции — простого голосования базовых алгоритмов $a_1, \ldots, a_T$:

$$a(x) = \arg\max_{y \in Y} \Gamma_y(x), \quad \Gamma_y(x_i) = \sum_{t=1}^{T} \bigl[a_t(x_i) = y\bigr].$$

тогда $M_i(a)$ — степень уверенности классификации $a(x_i)$.

1: $Z := X^\ell$;
2: **пока** $|Z| < \ell + k$
3: $\quad a := \mu(Z)$;
4: $\quad \Delta := \bigl\{x_i \in X^k \backslash Z \bigm| M_i(a) \geqslant M_0\bigr\}$;
5: $\quad y_i := a(x_i)$ для всех $x_i \in \Delta$;
6: $\quad Z := Z \cup \Delta$;

Кластеризация как задача дискретной оптимизации

Пусть $\rho(x, x')$ — *функция расстояния* между объектами.
Веса на парах объектов (близости): $w_{ij} = \exp(-\beta\rho(x_i, x_j))$,
где β — параметр.

Задача кластеризации:

$$\sum_{i=1}^{\ell+k}\sum_{j=i+1}^{\ell+k} w_{ij}\bigl[a_i \neq a_j\bigr] \to \min_{\{a_i \in Y\}}.$$

Задача частичного обучения:

$$\sum_{i=1}^{\ell+k}\sum_{j=i+1}^{\ell+k} w_{ij}\bigl[a_i \neq a_j\bigr] + \lambda\sum_{i=1}^{\ell}\bigl[a_i \neq y_i\bigr] \to \min_{\{a_i \in Y\}}.$$

где λ — ещё один параметр.

Алгоритм КНП: кластеризация

Графовый алгоритм КНП (кратчайший незамкнутый путь)

1: Найти пару вершин $(x_i, x_j) \in X^{\ell+k}$ с наименьшим $\rho(x_i, y_j)$ и соединить их ребром;
2: **пока** в выборке остаются изолированные точки
3: найти изолированную точку, ближайшую к некоторой неизолированной;
4: соединить эти две точки ребром;
5: удалить $K - 1$ самых длинных рёбер;

Задача частичного обучения: земенить только шаг 5...

Алгоритм КНП: частичное обучение

Графовый алгоритм КНП (кратчайший незамкнутый путь)

1: Найти пару вершин $(x_i, x_j) \in X^{\ell+k}$ с наименьшим $\rho(x_i, y_j)$ и соединить их ребром;
2: **пока** в выборке остаются изолированные точки
3: найти изолированную точку, ближайшую к некоторой неизолированной;
4: соединить эти две точки ребром;
5: ~~удалить $K - 1$ самых длинных рёбер;~~
6: **пока** есть путь между двумя вершинами разных классов
7: удалить самое длинное ребро на этом пути.

Метод k-средних: кластеризация

начальное приближение центров μ_y, $y \in Y$;
повторять
  E-шаг:
  отнести каждый x_i к ближайшему центру:
  $$y_i := \arg\min_{y \in Y} \rho(x_i, \mu_y), \quad i = 1, \dots, \ell + k;$$
  M-шаг:
  вычислить новые положения центров:
  $$\mu_y := \frac{\sum_{i=1}^{\ell+k} [y_i = y]\, x_i}{\sum_{i=1}^{\ell+k} [y_i = y]}, \quad \text{для всех } y \in Y;$$
пока y_i не перестанут изменяться;

Метод k-средних: частичное обучение

1: начальное приближение центров μ_y, $y \in Y$;

2: **повторять**

3: **Е-шаг:**

отнести каждый $x_i \in X^k$ к ближайшему центру:

$$y_i := \arg\min_{y \in Y} \rho(x_i, \mu_y), \quad i = \ell+1, \ldots, \ell+k;$$

4: **М-шаг:**

вычислить новые положения центров:

$$\mu_y := \frac{\sum_{i=1}^{\ell+k} [y_i = y]\, x_i}{\sum_{i=1}^{\ell+k} [y_i = y]}, \quad \text{для всех } y \in Y;$$

5: **пока** y_i не перестанут изменяться;

Алгоритм Ланса-Уильямса: кластеризация

Алгоритм иерархической кластеризации (Ланс, Уильямс, 1967)

1: $C_1 := \{\{x_1\}, \ldots, \{x_{\ell+k}\}\}$ — все кластеры 1-элементные;
$R_{\{x_i\}\{x_j\}} := \rho(x_i, x_j)$ — расстояния между ними;
2: **для всех** $t = 2, \ldots, \ell + k$ (t — номер итерации):
3: найти в C_{t-1} пару кластеров (U, V) с минимальным R_{UV};
4: слить их в один кластер:
$W := U \cup V$;
$C_t := C_{t-1} \cup \{W\} \setminus \{U, V\}$;
5: **для всех** $S \in C_t$
6: вычислить R_{WS} по формуле Ланса-Уильямса:
$R_{WS} := \alpha_U R_{US} + \alpha_V R_{VS} + \beta R_{UV} + \gamma \big| R_{US} - R_{VS} \big|$;

Алгоритм Ланса-Уильямса: частичное обучение

Алгоритм иерархической кластеризации (Ланс, Уильямс, 1967)

1: $C_1 := \bigl\{\{x_1\}, \dots, \{x_{\ell+k}\}\bigr\}$ — все кластеры 1-элементные;
 $R_{\{x_i\}\{x_j\}} := \rho(x_i, x_j)$ — расстояния между ними;
2: **для всех** $t = 2, \dots, \ell + k$ (t — номер итерации):
3: найти в C_{t-1} пару кластеров (U, V) с минимальным R_{UV},
 при условии, что в $U \cup V$ нет объектов с разными метками;
4: слить их в один кластер:
 $W := U \cup V$;
 $C_t := C_{t-1} \cup \{W\} \setminus \{U, V\}$;
5: **для всех** $S \in C_t$
6: вычислить R_{WS} по формуле Ланса-Уильямса:
 $R_{WS} := \alpha_U R_{US} + \alpha_V R_{VS} + \beta R_{UV} + \gamma \bigl|R_{US} - R_{VS}\bigr|$;

SVM: классификация

Линейный классификатор на два класса $Y = \{-1, 1\}$:

$$a(x) = \operatorname{sign}\bigl(\langle w, x\rangle - w_0\bigr), \quad w, x \in \mathbb{R}^n,\ w_0 \in \mathbb{R}.$$

Отступ объекта x_i:

$$M_i(w, w_0) = \bigl(\langle w, x_i\rangle - w_0\bigr) y_i.$$

Задача обучения весов w, w_0 по размеченной выборке:

$$Q(w, w_0) = \sum_{i=1}^{\ell} \bigl(1 - M_i(w, w_0)\bigr)_+ + \frac{1}{2C}\|w\|^2 \to \min_{w, w_0}.$$

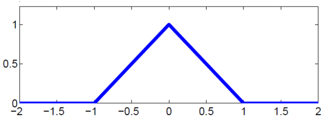
Функция $\mathscr{L}(M) = (1 - M)_+$ штрафует за уменьшение отступа.

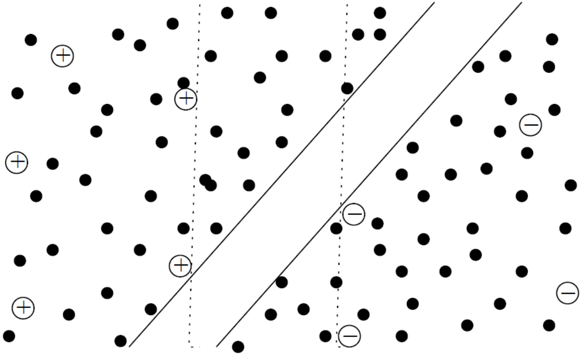
Идея!

Функция $\mathscr{L}(M) = \bigl(1 - |M|\bigr)_+$ штрафует за попадание объекта внутрь разделяющей полосы.

Функция потерь для трансдуктивного SVM

Функция потерь $\mathscr{L}(M) = (1 - |M|)_+$ штрафует за попадание объекта внутрь разделяющей полосы.





Transductive SVM: частичное обучение

Обучение весов w, w_0 по частично размеченной выборке:

$$Q(w, w_0) = \sum_{i=1}^{\ell} \bigl(1 - M_i(w, w_0)\bigr)_+ + \frac{1}{2C}\|w\|^2 + \\ + \gamma \sum_{i=\ell+1}^{\ell+k} \bigl(1 - |M_i(w, w_0)|\bigr)_+ \to \min_{w,w_0}.$$

Достоинства и недостатки TSVM:

- ⊕ как и в обычном SVM, можно использовать ядра;
- ⊕ имеются эффективные реализации для больших данных;
- ⊖ решение неустойчиво, если нет области разреженности;
- ⊖ требуется настройка двух параметров C, γ;

Sindhwani, Keerthi. Large scale semisupervised linear SVMs. SIGIR 2006.

Логистическая регрессия: классификация на 2 класса

Линейный классификатор на два класса $Y = \{-1, 1\}$:

$$a(x) = \operatorname{sign}\langle w, x\rangle, \quad x, w \in \mathbb{R}^n.$$

Вероятность того, что объект x_i относится к классу y:

$$P(y|x_i, w) = \frac{1}{1 + \exp(-\langle w, x_i\rangle y)}.$$

Задача максимизации регуляризованного правдоподобия:

$$Q(w) = \sum_{i=1}^{\ell} \log P(y_i|x_i, w) - \frac{1}{2C}\|w\|^2 \to \max_w,$$

Логистическая регрессия: классификация при произвольном Y

Линейный классификатор при произвольном числе классов $|Y|$:

$$a(x) = \arg\max_{y\in Y} \langle w_y, x\rangle, \quad x, w_y \in \mathbb{R}^n, \quad w \equiv (w_y)_{y\in Y}.$$

Вероятность того, что объект x_i относится к классу y:

$$P(y|x_i, w) = \frac{\exp\langle w_y, x_i\rangle}{\sum\limits_{c\in Y} \exp\langle w_c, x_i\rangle}.$$

Задача максимизации регуляризованного правдоподобия:

$$Q(w) = \sum_{i=1}^{\ell} \log P(y_i|x_i, w) - \frac{1}{2C}\|w\|^2 \to \max_w,$$

Логистическая регрессия: частичное обучение

Теперь учтём неразмеченные данные $X^k = \bigl\{x_{\ell+1}, \ldots, x_{\ell+k}\bigr\}$.
Пусть $b_j(x)$ — бинарные признаки, $j = 1, \ldots, m$.

Оценим вероятности $\mathsf{P}(y|b_j(x) = 1)$ двумя способами:

1) эмпирическая оценка по размеченным данным X^ℓ:

$$p_j(y) = \frac{\sum_{i=1}^{\ell} b_j(x_i)\bigl[y_i = y\bigr]}{\sum_{i=1}^{\ell} b_j(x_i)};$$

2) оценка по неразмеченным данным X^k и линейной модели w:

$$p_j(y, w) = \frac{\sum_{i=\ell+1}^{\ell+k} b_j(x_i) P(y|x_i, w)}{\sum_{i=\ell+1}^{\ell+k} b_j(x_i)}.$$

Будем минимизировать расстояние между $p_j(y)$ и $p_j(y, w)$.

Как оценить расстояние между двумя распределениями?

Пусть $p(y)$ и $q(y)$ — два дискретных распределения, $y \in Y$.

Проблема: малые различия в «хвостах» распределений могут приводить к существенным различиям статистических свойств.

$$E^2(p,q) = \sum_y \bigl(p(y) - q(y)\bigr)^2$$ — неадекватная мера расстояния;

$$X^2(p,q) = \sum_y \frac{\bigl(p(y) - q(y)\bigr)^2}{q(y)}$$ — статистика хи-квадрат;

$$H^2(p,q) = \frac{1}{2}\sum_y \bigl(\sqrt{p(y)} - \sqrt{q(y)}\bigr)^2$$ — расстояние Хелингера;

$$\mathrm{KL}(p\|q) = \sum_y p(y) \log \frac{p(y)}{q(y)}$$ — дивергенция Кульбака–Лейблера.

KL связана с *принципом максимума правдоподобия*

Свойства дивергенции Кульбака–Лейблера

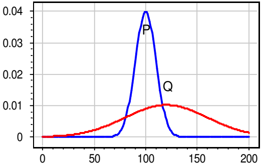
KL-дивергенция: $\mathrm{KL}(p\|q) = \sum\limits_y p(y)\log\dfrac{p(y)}{q(y)}$

1. $\mathrm{KL}(p\|q) \geqslant 0$; $\quad \mathrm{KL}(p\|q) = 0 \;\Leftrightarrow\; p = q$;

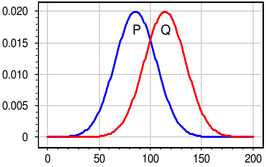
2. Если $p(y)$ — эмпирическое распределение, $q(y;\alpha)$ — параметрическая модель, то минимизация KL эквивалентна максимизации правдоподобия:

$$\mathrm{KL}(p\|q(\alpha)) = \sum_y p(y)\ln\frac{p(y)}{q(y;\alpha)} \to \min_\alpha \quad\Longleftrightarrow\quad \sum_y p(y)\ln q(y;\alpha) \to \max_\alpha.$$

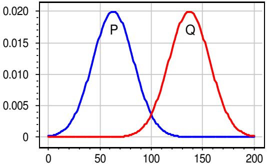
3. Если $\mathrm{KL}(p\|q) < \mathrm{KL}(q\|p)$, то p сильнее вложено в q, чем q в p:



$\mathrm{KL}(p\|q) = 0.442$
$\mathrm{KL}(q\|p) = 2.966$



$\mathrm{KL}(p\|q) = 0.444$
$\mathrm{KL}(q\|p) = 0.444$



$\mathrm{KL}(p\|q) = 2.969$
$\mathrm{KL}(q\|p) = 2.969$

Построение функционала качества

Кросс-энтропия — мера согласованности двух оценок, $p_j(y)$ и $p_j(y,w)$, одной и той же вероятности $\mathsf{P}(y|b_j(x)=1)$:

$$H_j(w) = \sum_{y\in Y} p_j(y)\log p_j(y,w) \to \max_w$$

(максимум достигается при $p_j(y) \equiv p_j(y,w)$).

Добавим суммарную согласованность по всем m признакам к функционалу регуляризованного правдоподобия:

$$Q(w) = \sum_{i=1}^{\ell} \log P(y_i|x_i,w) - \frac{1}{2C}\sum_{y\in Y}\|w_y\|^2 + \\ + \gamma \sum_{j=1}^{m}\sum_{y\in Y} p_j(y)\log\left(\frac{\sum_{i=\ell+1}^{\ell+k} b_j(x_i)P(y|x_i,w)}{\sum_{i=\ell+1}^{\ell+k} b_j(x_i)}\right) \to \max_w$$

Замечания про метод XR (Expectation Regularization)

1. Оптимизация $Q(w)$ — методом стохастического градиента.
2. Возможные варианты задания переменных b_j:
    1. $b_j(x) \equiv 1$, тогда $\mathsf{P}(y|b_j(x) = 1)$ — априорная вероятность класса y (label regularization) — хорошо подходит для задач с несбалансированными классами;
    2. $b_j(x) = \bigl[$термин j содержится в тексте $x\bigr]$ — для задач классификации и каталогизации текстов.
3. XR слабо чувствителен к выбору C и γ.
4. XR очень устойчив к погрешностям оценивания $p_j(y)$.
5. XR не требователен к числу размеченных объектов ℓ.
6. XR хорошо подходит для категоризации текстов.
7. XR показывает в экспериментах очень высокую точность.

Mann, McCallum. Simple, robust, scalable semi-supervised learning via expectation regularization. ICML 2007.

Резюме в конце лекции

- Задача SSL занимает промежуточное положение между классификацией и кластеризацией, но не сводится к ним.
- Простые методы-обёртки требуют многократного обучения, что вычислительно неэффективно.
- Методы кластеризации легко адаптируются к SSL путём введения ограничений (constrained clustering), но, как правило, вычислительно трудоёмки.
- Адаптация методов классификации реализуется сложнее, но приводит к более эффективным методам.
- Expectation Regularization — быстрый и точный метод, позволяющий учитывать дополнительную информацию.