Классификация с пересекающимися классами
НИС ФКН

Вахрамеева Елизавета, Полонская Диана

28 Ноября 2016

Постановка задачи

Формулировка

- Будем считать, что в задаче K классов:

$$y \in \{y_1, y_2 ... y_K\}$$

  каждый объект может относиться одновременно к нескольким классам
- Нужно найти наилучшую функцию(алгоритм) из пространства объектов в пространство ответов

$$a : X \to \{0,1\}^K$$

Постановка задачи

Пример

Такие задачи часто нужно решать в областях

- Обработки и анализа текстов
- Биоинформатики
- Компьютерного зрения

Метрики качества
Хэммингово расстояние

Z_i множество классов, к которым был отнесен объект
Y_i множество классов, к которым был на самом деле принадлежит объект

$$HammingLoss(a, X) = \frac{1}{m} \sum_{i=1}^{m} \frac{|Y_i \setminus Z_i| + |Z_i \setminus Y_i|}{K}$$

расстояние Хэмминга показывает долю классов, факт принадлежности которым угадан неверно

Метрики качества

Потери в рангах

Существуют алгоритмы, основанные на ранжировании классов для данного объекта так,
что меньший ранг класса означает меньшую вероятность принадлежности объекта к этому классу.

Y_i множество классов, к которым принадлежит x_i

$\bar{Y}_i$ множество классов, к которым не принадлежит x_i

Метрики качества

Потери в рангах

$$RankingLoss(a, X) =$$

$$= \frac{1}{m}\sum_{i=1}^{m}\frac{1}{|Y_i||\bar{Y}_i|}|\{(y', y'')|a(x_i, y') \leq a(x_i, y''), (y', y'') \in Y_i \times \bar{Y}_i\}|$$

показывает долю неправильно отранжированных пар классов

Метрики качества

Микро- и макро-усреднение

Обозначим за TP_j, FP_j, TN_j, FN_j значения TP, FP, TN, FN, посчитанные для $j-$ого класса
$B \in \{\text{Accuracy, Precision, Recall, F}\}$

- **микро-усреднение**

$$B_{micro}(a, X) = B(\frac{1}{m}\sum_{j=1}^{K} TP_j, \ldots, \frac{1}{m}\sum_{j=1}^{K} FN_j)$$

- **макро-усреднение**

$$B_{macro}(a, X) = \frac{1}{m}\sum_{j=1}^{K} B(TP_j, FP_j, TN_j, FN_j)$$

Микро- и макро-усреднение

Что выбрать?

Микро-усреднение

- чувствительно к размеру классов
- результаты на маленьком классе не заметны, оценка смещена в сторону популярных классов

Макро-усреднение

- не чувствительно к размеру классов
- каждый класс вносит одинаковый вклад

Обучение алгоритма

Два подхода

Методы сведения задачи к известной

- Независимая классификация (Binary Relevance)
- Цепочка классификаторов (Classifier Chains)
- Ранжирование классов (Calibrated Label Ranking)

Методы адаптации алгоритмов

- Метод k-ближайших соседей
- Решающее дерево (ML-DT)
- Ранговый метод опорных векторов (Rank-SVM)

Независимая классификация

Идея

Обучим K бинарных классификаторов

$$a_1(x) \dots a_K(x)$$

на выборках

$$(x_i, y_{i1})_{i=1}^m, \dots (x_i, y_{iK})_{i=1}^m$$

Тогда итоговый алгоритм:

$$a(x) = (a_1(x), \dots, a_K(x))$$

Независимая классификация

Достоинства и недостатки

Достоинства

- Очень простая идея сведения задачи к известной
- Может быть легко распараллелен

Недостатки

- Полностью игнорируются возможные связи между отдельными классами
- Метод чувствителен к классовому дисбалансу

Цепочка классификаторов

Идея

Обучение Зададим некую перестановку на множестве классов

$$\tau : \{1 \dots K\} \to \{1 \dots K\}$$

Построим цепочку бинарных классификаторов, где каждый последующий классификатор в цепочке учитывает ответы предыдущих

$$a_{\tau(j)}(x) \text{обучается на} \Big((x_i, y_{\tau(1)}, \dots y_{\tau(j-1)}), y_{\tau(j)} \Big)_{i=1}^{m}$$

Цепочка классификаторов

Идея

Предсказание для нового объекта

Пусть $c(x)$ итоговый алгоритм

$$c(x)_{\tau(j)}(x) = a_{\tau(j)}\big(x, a_{\tau(1)}(x), \ldots a_{\tau(j-1)}(x)\big)$$

Цепочка классификаторов

Идея

Проблема:
Выбор перестановки τ может сильно повлиять на результат.
Не понятно, как выбрать лучшую перестановку?
Решение:
Перебрать n случайных перестановок,
построить $C_1(x), \dots C_n(x)$ алгоритмов.

$C_i(x)$ будем обучать на сэмплированной с возвращением подвыборке X^i.
Рекомендуется брать $|X^i| = 0.67|X|$

Цепочка классификаторов

Идея

Объединение алгоритмов в композицию

$$C_j(x) \in \{0,1\}^K$$

$$W = \sum_{j=1}^{n} C_j(x) \in R^K$$

W_j сумма голосов за класс j

Нормируем вектор $W \to W^{norm}$

По заданному порогу t определяем, к каким классам отнесем данный объект

Цепочка классификаторов

Достоинства и недостатки подхода

Достоинства

- Сохраняется и используется информацию о возможной взаимосвязи между классами

Недостатки

- Не может быть распараллелен

Ранжирование классов

Идея

Добавим фиктивный класс в данные, а затем с помощью попарного сравнения(one-vs-one) для реальных классов и ove-vs-all сравнения для фиктивного класса отранжируем классы так, что ранги классов, к которым стоит отнести объект, будут всегда больше ранга фиктивного класса.

Ранжирование классов

Обучение

Обучим бинарные классификаторы $g_{js}(x)$ для всевозможных пар классов

$$(y_j, y_s)(1 \leq j \leq s \leq K)$$

.
Будем относить **x** к классу **j**, если $g_{js}(x) > 0$
и к классу **s** иначе

Ранжирование классов

Обучение

Каждый из алгоритмов $g_{js}(x)$ обучается по следующей выборке

$$X^{(js)} = \left(x_i, \psi(y_{ij}, y_{is})\right)_{i=1}^{i=m}$$

$$\psi(y_{ij}, y_{is}) = \begin{cases} +1, & y_{ij} = +1 \text{ and } y_{is} = -1 \\ -1, & y_{ij} = -1 \text{ and } y_{is} = +1 \end{cases}$$

Ранжирование классов
Обучение

Введем фиктивный класс λ и обучим еще K классификаторов

$$X^{(j\lambda)} = \left(x_i, \psi(y_{ij})\right)_{i=1}^{i=m}$$

$$\psi(y_{ij}, y_{is}) = \begin{cases} +1, & y_{ij} = +1 \\ -1, & \text{otherwise} \end{cases}$$

Ранжирование классов

Предсказание

Посчитаем голоса для каждого из класса на этом объекте:

$$C_j(x) = \sum_{s=1}^{j-1} [g_{sj} <= 0] + \sum_{s=j+1} K\,[g_{js} > 0]\, s$$

Добавим информацию от классификаторов, работающих с фиктивным классом:

$$C_j^*(x) = C_j(x) + [g_{s\lambda} > 0]$$

Голоса для фиктивного класса:

$$C_\lambda^*(x) = \sum_{j=1}^{K} [g_{j\lambda} \leq 0]$$

Ранжирование классов

Предсказание

Множество классов, к которым принадлежит данный объект определяется как:

$$\{y_j | C_j^*(x) > C_\lambda^*(x), 1 \leq j \leq K\}$$

Достоинства

- задача сводится к one-vs-one классификации, что помогает бороться с эффектом дисбаланса в размере классов

Недостатки

- Сложность становиться квадратичной относительно количества классов

Метод k-ближайших соседей (ML-kNN)

Идея

Адаптируем метод k-ближайших соседей с Евклидовой метрикой, используя правило MAP (maximum a posteriori probability). Будем делать предсказания на основе информации о классах соседей.

Метод k-ближайших соседей (ML-kNN)

Обучение

$N(x)$ – множество k-ближайших соседей в тестовой выборке
Для j-ого признака считаем следующую статистику:

$$C_j = \sum_{(x^*, Y^*) \in N(x)} [\![y_j \in Y^*]\!]$$

$H_j = \{x \text{ имеет признак } y_j\}$
$Y = \{y_j | \mathbb{P}(H_j|C_j)/\mathbb{P}(\neg H_j|C_j) >, 1 \leq j \leq K\}$
По теореме Байеса:

$$\frac{\mathbb{P}(H_j|C_j)}{\mathbb{P}(\neg H_j|C_j)} = \frac{\mathbb{P}(H_j) \cdot \mathbb{P}(C_j|H_j)}{\mathbb{P}(\neg H_j) \cdot \mathbb{P}(C_j|\neg H_j)}$$

Метод k-ближайших соседей (ML-kNN)

Обучение

Априорная вероятность:

$$\mathbb{P}(H_j) = \frac{s + \sum_{i=1}^{m} [\![y_j \in Y_i]\!]}{s \cdot 2 + m}$$

Массивы частот:

$$\mathcal{K}_j[r] = \sum_{i=1}^{m} [\![y_j \in Y_i]\!] \cdot [\![\delta_j(x_i) = r]\!], (0 \leq r \leq k)$$

$$\bar{\mathcal{K}}_j[r] = \sum_{i=1}^{m} [\![y_j \notin Y_i]\!] \cdot [\![\delta_j(x_i) = r]\!], (0 \leq r \leq k)$$

$$\delta_j(x_i) = \sum_{(x^*, Y^*) \in N(x_i)} [\![y_j \in Y^*]\!]$$

Правдоподобия:

$$\mathbb{P}(C_j | H_j) = \frac{s + \mathcal{K}_j[C_j]}{s \cdot (k+1) + \sum_{r=0}^{k} \bar{\mathcal{K}}_j[r]}$$

Метод k-ближайших соседей (ML-kNN)

Достоинства и недостатки

Достоинства

- Совмещает в себе lasy learning и Байесовский вывод
- Проблема классового дисбаланса смягчается благодаря априорным вероятностям

Недостатки

- Не учитывает корреляцию между признаками
- Если число признаков велико, то при Евклидовой метрике выбор k ближайших соседей становится практически произвольным.

Решающее дерево (ML-DT)

Идея

Основная идея этого алгоритма заключается в обобщении метода решающих деревьев. Будем использовать критерий информативности, основанный на многоклассовой энтропии, а дерево строить рекурсивно.

Решающее дерево (ML-DT)

Обучение

$\tau = \{(x_i, Y_i) | 1 \leq i \leq n\}$ – выборка

$$IG(\tau, l, \nu) = MLEnt(\tau) - \sum_{p \in \{-,+\}} \frac{|\tau^p|}{|\tau|} \cdot MLEnt(\tau^p)$$

Два подхода для подсчета MLEnt:

$$\widehat{MLEnt}(\tau) = - \sum_{Y \subseteq \mathcal{Y}} \mathbb{P}(Y) \cdot \log_2(\mathbb{P}(Y)),$$

где $\mathbb{P}(Y) = \frac{\sum_i^n [\![Y_i = Y]\!]}{n}$

$$MLEnt(\tau) = \sum_{j=1}^{K} -p_j - \log_2 pj - (1 - pj) \log_2(1 - p_j),$$

где $p_j = \frac{\sum_i^n [\![y_j \in Y_i]\!]}{n}$

Решающее дерево (ML-DT)

Предсказание

Множество классов, к которым принадлежит данный объект определяется как:

$$Y = \{y_j | p_j > 0.5, 1 \leq j \leq K\}$$

Решающее дерево (ML-DT)

Предсказание

Достоинства

- Эффективная адаптация алгоритма к данным

Недостатки

- Не учитывает корреляцию между признаками при подсчете энтропии

Улучшения: можно делать стрижку, составлять композиции

Ранговый метод опорных векторов (Rank-SVM)

Идея

Следует адаптировать стратегию максимизации отступа. Множество линейных классификаторов, в свою очередь, должны минимизировать эмпирические ранговые потери.

Ранговый метод опорных векторов (Rank-SVM)

Обучение

Множество классификаторов:

$$\mathcal{W} = \{(w_j, b_j) | 1 \leq j \leq K\}$$

Отступ системы на (x_i, Y_i):

$$\min_{(y_j, y_k) \in Y_i \times \bar{Y}_i} \frac{\langle w_j - w_k, x_i \rangle + b_j - b_k}{||w_j - w_k||}$$

Отступ для всей обучающей выборки:

$$\min_{(x_i, Y_i) \in \mathcal{D}} \min_{(y_j, y_k) \in Y_i \times \bar{Y}_i} \frac{\langle w_j - w_k, x_i \rangle + b_j - b_k}{||w_j - w_k||}$$

Ранговый метод опорных векторов (Rank-SVM)

Обучение

Разделимый случай

После нормировки:

$$\max_{\mathcal{W}} \min_{(x_i, Y_i) \in \mathcal{D}} \min_{(y_j, y_k) \in Y_i \times \bar{Y}_i} \frac{1}{||w_j - w_k||}$$

$$\min_{\mathcal{W}} \max_{1 \leq j \leq k \leq K} ||w_j - w_k||$$

$$\langle w_j - w_k, x_i \rangle + b_j - b_k \geq 1, (1 \leq i \leq m)$$

И еще раз упростим:

$$\min_{\mathcal{W}} \sum_{j=1}^{K} ||w_j||^2$$

$$\langle w_j - w_k, x_i \rangle + b_j - b_k \geq 1, (1 \leq i \leq m)$$

Ранговый метод опорных векторов (Rank-SVM)

Обучение

Неразделимый случай

$$\min_{\{\mathcal{W},\Xi\}} \sum_{j=1}^{K} ||w_j||^2 + C \sum_{i=1}^{m} \frac{1}{|Y_i||\bar{Y}_i|} \sum_{(y_j,y_k)\in Y_i\times\bar{Y}_i} \xi_{ijk}$$

$$\langle w_j - w_k, x_i \rangle + b_j - b_k \geq 1 - \xi_{ijk}, (1 \leq i \leq m)$$

$$\xi ijk \geq 0, (1 \leq i \leq m, (y_j, y_k \in Y_i \times \bar{Y}_i))$$

Ранговый метод опорных векторов (Rank-SVM)

Предсказание

Зададим пороговую функцию $t(\cdot)$:
$t(x) = \langle w^*, f^*(x) \rangle + b^*$, где $f^*(x) = (f(x,y_1), \cdots f(x,y_K))^T$,
$f(x, y_j) = \langle w_j, x \rangle + b_j$
Множество классов, к которым принадлежит данный объект определяется как:

$$Y = \{y_j | \langle w_j, x \rangle + b_j > t(x), 1 \leq j \leq K\}$$

Ранговый метод опорных векторов (Rank-SVM)

Достоинства

- Учитывает корреляцию между классами и строит отступ для релевантных-нерелевантных пар

Недостатки

- Требует больших затрат памяти

Сравнение характеристик

Algorithm	Basic Idea	Order of Correlations	Complexity [Train/Test]	Tested Domains	Optimized Metric
Binary Relevance [5]	Fit multi-label data to q binary classifiers	first-order	$\mathcal{O}(q \cdot \mathcal{F}_{\mathcal{B}}(m,d))/$ $\mathcal{O}(q \cdot \mathcal{F}'_{\mathcal{B}}(d))$	image	classification (hamming loss)
Classifier Chains [72]	Fit multi-label data to a chain of binary classifiers	high-order	$\mathcal{O}(q \cdot \mathcal{F}_{\mathcal{B}}(m,d+q))/$ $\mathcal{O}(q \cdot \mathcal{F}'_{\mathcal{B}}(d+q))$	image, video text, biology	classification (hamming loss)
Calibrated Label Ranking [30]	Fit multi-label data to $\frac{q(q+1)}{2}$ binary classifiers	second-order	$\mathcal{O}(q^2 \cdot \mathcal{F}_{\mathcal{B}}(m,d))/$ $\mathcal{O}(q^2 \cdot \mathcal{F}'_{\mathcal{B}}(d))$	image, text biology	Ranking (ranking loss)
ML-kNN [108]	Fit k-nearest neighbor to multi-label data	first-order	$\mathcal{O}(m^2 d + qmk)/$ $\mathcal{O}(md + qk)$	image, text biology	classification (hamming loss)
ML-DT [16]	Fit decision tree to multi-label data	first-order	$\mathcal{O}(mdq)/\mathcal{O}(mq)$	biology	classification (hamming loss)
Rank-SVM [27]	Fit kernel learning to multi-label data	second-order	$\mathcal{O}(\mathcal{F}_{\mathrm{QP}}(dq + mq^2, mq^2)$ $+q^2(q+m))/\mathcal{O}(dq)$	biology	Ranking (ranking loss)

Другие области применения

- Классификация потоков данных
- Порядковая классификация
- Многовариантное обучение

Другие области применения

Классификация потоков данных

- Новости, emails, микроблоги, пр.
- Concept drift problem

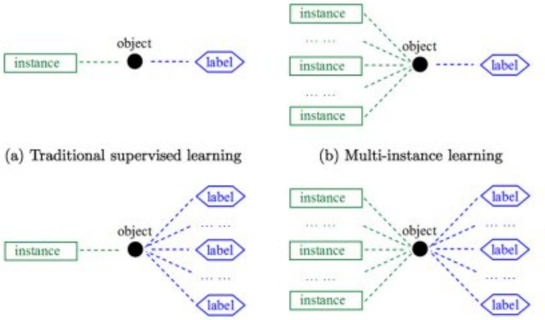
Другие области применения

Порядковая классификация

- Часто используется в социологии
- Можно свести к композиции мультиклассовых
- Применим Rank-SVM

Другие области применения

Многовариантное обучение



(a) Traditional supervised learning (b) Multi-instance learning

(c) Multi-label learning (d) Multi-instance multi-label learning