Машинное обучение. Multiclass, Ranking and Complex Prediction Problems

Алексей Колесов

Белорусский государственный университет

1 ноября 2017 г.

Содержание

Классификация на несколько классов

- ищем $h : X \to Y$, где Y — конечное число классов
- считаем, что $Y = \{1, \dots, k\}$
- примеры: классификация документам по темам, определение породы котика на картинке

One-vs-all

- тренируем k классификаторов: $h_i : X \to \{-1, 1\}$
- $S = (x_1, y_1), \ldots, (x_m, y_m) \to$
  $S_i = (x_1, (-1)^{1_{y_1 \neq i}}), \ldots, (x_m, (-1)^{1_{y_m \neq i}})$
- итоговый классификатор: $h(x) \in \operatorname{argmax}_{i \in [k]} h_i(x)$

Проблемы one-vs-all

- как выбрать, если несколько $h_i(x) = 1$?
- например, можно выбирать минимальный индекс
- лучше, если h_i имеет смысл «уверенности»
- например, в линейных: $h(x) \in \operatorname{argmax}_{i \in [k]} \langle w_i, x \rangle$

Алгоритм

Алгоритм 1 One-vs-All

Вход: $S = ((x_1, y_1), \dots, (x_m, y_m))$; $y_i \in [k]$

Вход: алгоритм бинарной классификации A

1: **for** $i = 1, \dots, k$ **do**

2: $\quad S_i = (x_1, (-1)^{1_{y_1 \neq i}}), \dots, (x_m, (-1)^{1_{y_m \neq i}})$

3: $\quad h_i = A(S_i)$

4: **end for**

5: **return** $h(x) \in \operatorname{argmax}_{i \in [k]} h_i(x)$

All-Pairs

- тренируем C_k^2 классификаторов: $h_{i,j} : X \to \{-1, 1\}$ (принадлежит классу i или j)
- $S = (x_1, y_1), \ldots, (x_m, y_m) \to$ $S_{i,j} = \{(x, 1) : (x, i) \in S\} \cup \{(x, -1) : (x, j) \in S\}$
- выбираем тот, у которого больше всего побед

Алгоритм

Алгоритм 2 All-Pairs

Вход: $S = ((x_1, y_1), \ldots, (x_m, y_m))$; $y_i \in [k]$

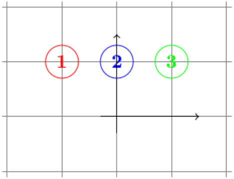
Вход: алгоритм бинарной классификации A

1: **for** $i = 1, \ldots, k-1$ **do**
2:     **for** $j = i+1, \ldots, k$ **do**
3:         $S_{i,j} = \{(x, 1) : (x, i) \in S\} \cup \{(x, -1) : (x, j) \in S\}$
4:         $h_{i,j} = A(S_{i,j})$
5:     **end for**
6: **end for**
7: **return** $h(x) \in \mathrm{argmax}_{i \in [k]} \left(\sum_{j \in [k]} \mathrm{sign}(j - i) h_{\min(i,j), \max(i,j)}(x) \right)$

Выводы

- сведение к бинарной классификации — простой способ решить задачу
- проблема: **оптимизируем не то поведение, что используем**

Пример



- пусть $P[y=1] = P[y=3] = 40\%$, $P[y=2] = 20\%$
- в `halfspaces` лучшее решение в one-vs-all для класса 2: $h(x) \equiv -1$
- $w_1 = (-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$, $w_2 = (0, 1)$, $w_1 = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ не ошибается

Непосредственные методы мультиклассовой классификации

- постараемся найти непосредственный метод решения задачи
- обобщим бинарную классификацию
- заметим, что

$$h(x) = \operatorname{sign}(\langle w, x \rangle) \Leftrightarrow h(x) = \operatorname*{argmax}_{y \in \{-1,1\}} \langle w, yx \rangle$$

Class-sensitive feature mapping

- зададимся $\Psi : X \times Y \to \mathbb{R}^d$ (class-sensitive feature mapping)
- $\Psi(x, y)$ **можно понимать**, как меру того, насколько объекту x подходит класс y
- $h(x) = \underset{y \in [k]}{\operatorname{argmax}} \langle w, \Psi(x, y) \rangle$
- $H_{\Psi, W} = \{x \mapsto \underset{y \in [k]}{\operatorname{argmax}} \langle w, \Psi(x, y) \rangle : w \in W\}$ задаёт класс гипотез

Как выбрать Ψ

- выбор Ψ критически важен для решения задачи
- можно строить независимые от задачи конструкции (multivector construction)
- можно пытаться добавить inductive bias (tf-idf)

The Multivector construction

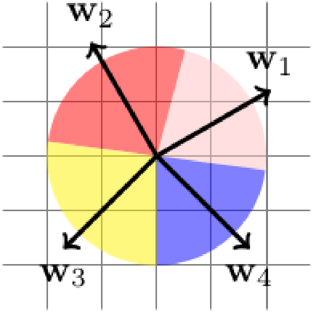
- пусть $X = \mathbb{R}^n$
- зададим $\Psi : X \times Y \to \mathbb{R}^d$, $d = nk$

$$\Psi(x, y) = \left[\underbrace{0, \ldots, 0}_{\in \mathbb{R}^{(y-1)n}}, \underbrace{x_1, \ldots, x_n}_{\in \mathbb{R}^n}, \underbrace{0, \ldots, 0}_{\in \mathbb{R}^{(k-y)n}} \right]$$

- $\langle w, \Psi(x, y) \rangle = \langle w_y, x \rangle$, где $w = [w_1, \ldots, w_k]$
- решающее правило:

$$h(x, y) = \underset{y \in Y}{\operatorname{argmax}} \langle w_y, x \rangle$$

Геометрическая интерпретация



TF-IDF

- пусть X — набор документов, Y — набор тем, d — количество слов в языке, m — количество документов
- $\mathrm{TF}(j, x)$ — количество раз, когда слово j встречалось в документе x (term frequency)
- $\mathrm{DF}(j, y)$ — количество раз, когда в документе с темой **не** y встречалось слово j
- следующая величина называется `TF-IDF` (term-frequency-inverse-document-frequency):

$$\Psi_j(x, y) = \mathrm{TF}(j, x) \log \left(\frac{m}{\mathrm{DF}(j, y)} \right)$$

TF-IDF: замечания

- `TF-IDF` велик, когда слово нечасто встречается в других темах, но часто в этом документе
- в отличие от multivector-construction размерность $\Psi(x, y)$ не зависит от $|Y|$

Cost-sensitive classification

- не все ошибки одинаково полезны
- введём $\Delta : Y \times Y \to \mathbb{R}_+$ — функция потерь, когда предсказали y', а правильный ответ y: $\Delta(y', y)$
- $\Delta(y, y) = 0$
- $0 - 1$-loss: $\Delta(y', y) = 1_{y' \neq y}$

ERM для multiclass

- есть $H_{\Psi,W}$, Δ
- можем минимизировать эмпирический риск:
$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m} \Delta(h(x_i), y_i)$$
- если $W = \mathbb{R}^d$ и **выполняется гипотеза реализуемости**, то есть эффективное решение

Сведение к линейному программированию

- в случае гипотезы реализуемости мы должны найти $w \in \mathbb{R}^d$:
$$\forall i \in [m],\ y_i = \operatorname*{argmax}_{y \in Y} \langle w, \Psi(x_i, y) \rangle$$
- эквивалентно:
$$\forall i \in [m], \forall y \in Y \setminus \{y_i\}\ \langle w, \Psi(x_i, y_i) \rangle > \langle w, \Psi(x_i, y) \rangle$$
- таким образом, можно решать с помощью фреймворка линейного программирования
- есть обобщение алгоритма batch perceptron

hinge-loss

- $\max\{0, 1 - y\langle w, x\rangle\}$ — `hinge-loss` для бинарной классификации
- попытаемся обобщить для $h_w(x) = \operatorname*{argmax}_{y' \in Y} \langle w, \Psi(x, y')\rangle$
- необходимо, чтоб новый лосс был оценкой сверху для оригинального $(\Delta(h_w(x), y))$

Обобщённый hinge-loss

Имеем:

$$h_w(x) = \operatorname*{argmax}_{y' \in Y} \langle w, \Psi(x, y') \rangle$$

Распишем:

$$\begin{align}
\langle w, \Psi(x, y) \rangle &\leqslant \langle w, \Psi(x, h_w(x)) \rangle \tag{1}\\
\Delta(h_w(x), y) &\leqslant \Delta(h_w(x), y) + \langle w, \Psi(x, h_w(x)) - \Psi(x, y) \rangle \tag{2}\\
\Delta(h_w(x), y) &\leqslant \max_{y' \in Y} (\Delta(y', y) + \langle w, \Psi(x, y') - \Psi(x, y) \rangle) \tag{3}
\end{align}$$

Величина $l(w, (x, y)) = \max_{y' \in Y} (\Delta(y', y) + \langle w, \Psi(x, y') - \Psi(x, y) \rangle)$ называется обобщённым hinge-loss-м

Обобщённый hinge-loss: свойства

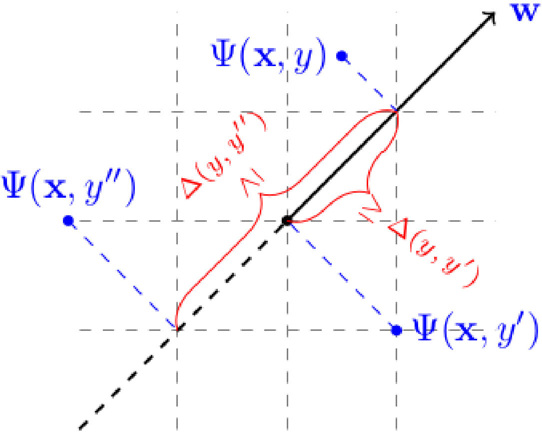
$$l(w,(x,y)) = \max_{y' \in Y}(\Delta(y',y) + \langle w, \Psi(x,y') - \Psi(x,y)\rangle)$$

- $l(w,(x,y)) = \Delta(y',y)$, если «скор» правильной метки больше любой остальной хотя бы на $\Delta(y',y)$
- функция является выпуклой
- является ρ-липшицевой для $\rho = \max_{y' \in Y} ||\Psi(x,y') - \Psi(x,y)||$

Геометрическая интуиция

- Ψ отображает каждый x в $|Y|$ векторов в $\mathbb{R}^d$
- $l(w,(x,y))=0$, если найдётся w, что проекции $\Psi(x,y_i)$ на вектор (т.е. скаляры $\langle w,\Psi(x,y_i)\rangle$) обладают следующими свойствами:
    - у точки с правильной меткой y самый большой скаляр
    - у всех $y_i\neq y$ скаляр меньше хотя бы на $\Delta(y,y_i)$; величина $\langle w,\Psi(x,y)\rangle-\langle w,\Psi(x,y_i)\rangle$ называют **отступом** (margin)

Геометрическая интуиция



Multiclass SVM

Multiclass SVM заключается в оптимизации:

$$\min_{w \in \mathbb{R}^d} \left(\lambda ||w||^2 + \frac{1}{m} \sum_{i=1}^{m} \max_{y' \in Y} (\Delta(y', y) + \langle w, \Psi(x, y') - \Psi(x, y) \rangle) \right)$$

Гарантии для Multiclass SVM

Пусть $||\Psi(x, y)|| \leqslant \rho/2$, $B > 0$. Тогда если мы решим задачу Multiclass SVM с $\lambda = \sqrt{\frac{2\rho^2}{B^2 m}}$, то:

$$\mathop{\mathbb{E}}_{S \sim D^m} [L_D^{\Delta}(h_w)] \leqslant \min_{u : ||u|| \leqslant B} L_D^{g-hinge}(u) + \sqrt{\frac{8\rho^2 B^2}{m}}$$

Выпуклая оптимизация для multiclass

- для SGD можно получить похожую оценкую
- они не зависят от $|Y|$ напрямую
- при большом $|Y|$ сложно найти u с $||u|| \leqslant B$, для которых $L_D^{g-hinge}$ небольшое

Содержание

Предсказание структурированного вывода

- в задачах типа «предсказание структурированного вывода» $|Y|$ очень велико, но Y организован в некоторую структуру
- например, распознавание напечатанного слова (OCR — optical character recognition)
- пусть мы умеем, сегментировать изображение
- X — множество последовательностей картинок, Y — множество последовательностей букв
- $|Y|$ растёт экспоненциально с длиной слова

Предсказание структурированного вывода

- попробуем решить задачу с помощью линейных классификаторов
- нужно найти разумную Δ и Ψ (для которой маленький approximation error)
- можем использовать SGD

Проблемы большого $|Y|$

- чтоб решить задачу мультиклассовой классификации нужно сделать максимизацию по большому Y. Как эффективно вычислять?
- как эффективно тренировать w?
- как избежать переобучения?

Предсказание структурированного вывода

- гарантии не зависят от $|Y|$, переобучение — не проблема
- для борьбы с вычислительной сложностью выберем Δ и Ψ таким образом, что максимизацию можно делать эффективно
- пусть максимальная длина слова r, размер алфавита q, $\Delta(y, y') = \frac{1}{r}\sum_{i=1}^{r} 1_{y_i \neq y'_i}$

Вариант Ψ

- будем считать, что $x \in \mathbb{R}^{n \times r}$ — r изображений по n пикселей
- $x_{i,j}$ — i-й пиксель в j-м изображении (например, оттенок серого)
- размер Ψ будет $nq + q^2$
- признаки первого типа: $\Psi_{i,j,1}(x, y) = \frac{1}{r} \sum_{t=1}^{r} x_{i,t} 1_{y_t = j}$
- признаки второго типа: $\Psi_{i,j,2}(x, y) = \frac{1}{r} \sum_{t=2}^{r} 1_{y_{t-1} = j} 1_{y_t = i}$
- $h_w(x) = \mathrm{argmax}_{y \in Y} \langle w, \Psi(x, y) \rangle$

Вычисление Ψ

$$h_w(x) = \mathrm{argmax}_{y \in Y} \langle w, \Psi(x, y) \rangle$$

- вычислять h_w до сих пор экспоненциально сложно
- заметим, что максимизировать величину можно с помощью динамического программирования
- $\Psi(x, y)$ допускает представление в виде $\sum_{t=1}^{r} \psi(x, y_t, y_{t-1})$
- $\psi_{i,j,1}(x, y_t, y_{t-1}) = x_{i,t} 1_{y_t = j}$
- $\psi_{i,j,2}(x, y_t, y_{t-1}) = 1_{y_t = i} 1_{y_{t-1} = j}$
- $h_w(x) = \operatorname*{argmax}_{y \in Y} \sum_{t=1}^{r} \langle w, \psi(x, y_t, y_{t-1}) \rangle$

Содержание

Ранжирование

- ранжирование — задача упорядочивания объектов по их «релевантности»
- пример: упорядочивание ответов поисковой системы на запрос, упорядочивание транзакций кредитных карт по подозрительности
- $X^* = \bigcup_{n=1}^{\infty} X^n$
- h принимает $\overline{x} = (x_1, \ldots, x_r) \in X^*$ и возвращает перестановку $[r]$
- обычно h возвращает вектор y, сортировка которого индуцирует перестановку $\pi(y)$

Пример нотации $\pi(y)$

- пусть $y = (2, 1, 6, -1, -0.5)$
- тогда $\pi(y) = (4, 3, 5, 1, 2)$
- $\pi_i(y)$ — место в отсортированном порядке у объекта x_i (например, $\pi_2(y) = 3$ в нашем примере)

Функции потерь

- в PAC-модели $Z = \bigcup_{r=1}^{\infty}(X^r \times \mathbb{R}^r)$
- будем искать $l(h, (\overline{x}, y)) = \Delta(h(\overline{x}), y)$:
  - $0-1$-loss: равен нулю, если $y' = y$ и единице иначе
  - функция потерь Кендалла-Тау:

$$\Delta(y', y) = \frac{2}{r(r-1)} \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} 1_{\text{sign}(y'_i - y'_j) \neq \text{sign}(y_i - y_j)}$$

  - `Normalized discounted cummulative gain` (NDCG): зададимся неубывающей функцией $D : \mathbb{N} \to R_+$. Обозначим:

$$G(y', y) = \sum_{i=1}^{r} D(\pi(y')_i) y_i$$

Тогда

$$\Delta(y', y) = 1 - \frac{G(y', y)}{G(y, y)} = \frac{1}{G(y, y)} \sum_{i=1}^{r} \left(D(\pi_i(y) - \pi_i(y')\right) y_i$$

Выпуклая оптимизация для функций потерь в задачах ранжирования

- один из естественных подходов — ранжировать объекты в зависимости от $\langle w, x_i \rangle$ для некоторого w
- $h_w((x_1, \ldots, x_r)) = (\langle w, x_1 \rangle, \ldots, \langle w, x_r \rangle)$
- необходимо вывести **эффективно вычислимую** суррогатную функцию потерь
- тогда можно применять ERM

Hinge loss для Kendell-Tau

- `Kendell-Tau` функция потерь — среднее $0-1$ функций потерь:

$$1_{\mathrm{sign}(y'_i-y'_j)\neq\mathrm{sign}(y_i-y_j)} = 1_{\mathrm{sign}(y_i-y_j)(y'_i-y'_j)\leqslant 0}$$

- $y'_i - y'_j = \langle w, x_i - x_j\rangle$
- $1_{\mathrm{sign}(y_i-y_j)(y'_i-y'_j)\leqslant 0} \leqslant \max\{0, 1-\mathrm{sign}(y_i-y_j)\langle w, x_i-x_j\rangle\}$

$$\Delta(h_w(\overline{x}), y) \leqslant$$
$$\frac{2}{r(r-1)}\sum_{i=1}^{r-1}\sum_{j=i+1}^{r}\max\{0, 1-\mathrm{sign}(y_i-y_j)\langle w, x_i-x_j\rangle\}$$

Алгоритмы learning to rank

- pointwise
- pairwise
- listwise

Bipartite ranking

- для ранжирования мы использовали y, который «задавали порядок» объектов
- порядок может быть частичным
- например, $y = \{-1; +1\}^r$ (`bipartite ranking`)

Bipartite ranking vs бинарная классификация

- рассмотрим задачу определения мошенничества в транзакциях с кредитным картами
- можно предположить, что 99.9% транзакций — хорошие
- $h(x) \equiv 0$ достигнет ошибки в 0.1% — ошибка маленькая, но классификатор бессмысленный
- нужно выбрать более полезную функцию потерь

Многомерные показатели эффективности

Многие классификаторы можно представить в виде

$$h(x) = \mathrm{sign}(y'_i - \theta)$$

Определим:

$$\begin{aligned}
\text{True positives:}\quad & \mathrm{TP} = |\{i : y_i = +1 \wedge \mathrm{sign}(y'_i - \theta) = +1\}| \\
\text{False positives:}\quad & \mathrm{FP} = |\{i : y_i = -1 \wedge \mathrm{sign}(y'_i - \theta) = +1\}| \\
\text{False negatives:}\quad & \mathrm{FN} = |\{i : y_i = +1 \wedge \mathrm{sign}(y'_i - \theta) = -1\}| \\
\text{True negatives:}\quad & \mathrm{TN} = |\{i : y_i = -1 \wedge \mathrm{sign}(y'_i - \theta) = -1\}|
\end{aligned}$$

Многомерные показатели эффективности

- **точность или чувствительность** (precision, sensitivity): $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$
- **специфичность** (specificity): $\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}$
- **полнота** (recall): $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$

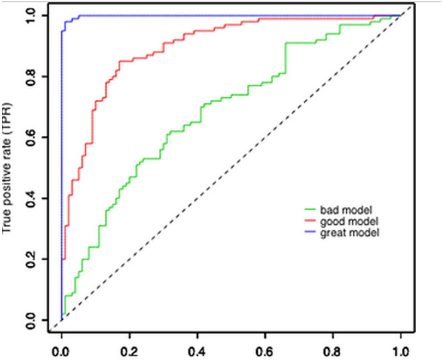
При увеличении θ **обычно** полнота падает, зато растёт точность и специфичность

Многомерные показатели эффективности

- усреднённая чувствительность и специфичность: $\frac{1}{2}\left(\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}+\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}\right)$
- F_1-score: гармоническое среднее между precision и recall:
$$F_1=\frac{2}{\frac{1}{\text{precision}}+\frac{1}{\text{recall}}}=\frac{2\,\mathrm{TP}}{2\,\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$$
- F_β-score: F_1, но `recall` в β^2 раз важней precision
- recall@k: лучшая полнота, если содержим как максимум k положительных примеров
- precision@k: лучшая точность, если содержим как минимум k положительных примеров

ROC-AUC

Часто строят график чувствительности (TPR) от (FPR $= 1-$ специфичность) — ROC-кривая (receiver operating characteristic)



ROC-AUC

Площадь под этим графиком называется ROC-AUC (area under curve):

Физический смысл: вероятность того, что положительный объект будет ранжирован выше отрицательного

Содержание

Итоги

- рассмотрели задачу классификации на много классов (сведение к бинарной и собственные методы)
- рассмотрели задачу, где выход структурирован
- рассмотрели задачу ранжирования

Литература

- Shai Shalev-Shwartz and Shai Ben-David — Understanding Machine Learning: From theory to algorithms (глава 17)