композиции классификаторов

Евгений Борисов

композиции классификаторов

методы ML

- *метрические* - померять расстояния, определить ближайших
- *логические* - построить правило (комбинацию предикатов)
- *линейные* - построить разделяющую поверхность
- *статистические* - восстановить плотность, определить вероятность
- ***композиции*** - собрать несколько классификаторов в один

композиции классификаторов

X - объекты   y - метки

$a(x)=C(b(x))$ - классификатор

$b : X \to R$ - базовый алгоритм
$C : R \to y$ - решающее правило

R - множество оценок (существенно шире чем y)

композиции классификаторов

примеры оценок и решающих правил

$a(x) = \mathrm{sign}(b(x))$ - **классификация на 2 класса**

$y=\{-1,0,1\}$ - метки

$b : X \to \mathbb{R}$ - базовый алгоритм

$C(b) = \mathrm{sign}(b)$ - решающее правило

композиции классификаторов

примеры оценок и решающих правил

$a(x) = \mathrm{sign}(b(x))$ - **классификация на 2 класса**

$y=\{-1,0,1\}$ - метки
$b : X \to \mathbb{R}$ - базовый алгоритм
$C(b) = \mathrm{sign}(b)$ - решающее правило

$a(x)=\mathrm{argmax}\ b_t(x)$ - **классификация на m классов**

$y=\{1,..,m\}$ - метки
$R=\mathbb{R}^m$ - оценки
$b : X \to \mathbb{R}^m$ - базовый алгоритм
$C(b_1,...b_m)=\mathrm{argmax}\ b_t$ - оценки

композиции классификаторов

примеры оценок и решающих правил

$a(x) = \mathrm{sign}(b(x))$ - **классификация на 2 класса**

$y=\{-1,0,1\}$ - метки
$b : X \to \mathbb{R}$ - базовый алгоритм
$C(b) = \mathrm{sign}(b)$ - решающее правило

$a(x)=\mathrm{argmax}\ b_t(x)$ - **классификация на m классов**

$y=\{1,..,m\}$ - метки
$R=\mathbb{R}^m$ - оценки
$b : X \to \mathbb{R}^m$ - базовый алгоритм
$C(b_1,...b_m)=\mathrm{argmax}\ b_t$ - оценки

$a(x)=b(x)$ - **регрессия**

$y=R=\mathbb{R}$ - метки / оценки
$b : X \to \mathbb{R}$ - базовый алгоритм
$C(b)=b$ - решающее правило (вырождено)

композиции классификаторов

<u>Идея:</u> из нескольких «плохих» классификаторов
собрать один «хороший»

композиции классификаторов

Идея: из нескольких «плохих» классификаторов собрать один «хороший»

X - объекты y - метки

$b_t : X \to R$ - базовый алгоритм
$C : R \to y$ - решающее правило
R - множество оценок

композиция базовых алгоритмов $b_t(x)$

$a(x) = C(F(b_1(x),...,b_T(x)))$

$F : R^T \to R$ - корректирующая операция

композиции классификаторов

$$a(x)=C\left(F\left(b_1(x),\ldots b_T(x)\right)\right)$$ композиция из классификаторов b_t

примеры корректирующих операций

$$F\left(b_1(x),\ldots b_T(x)\right)=\frac{1}{T}\sum_{t=1}^{T} b_t(x)$$ простое голосование

композиции классификаторов

$a(x)=C\left(F\left(b_1(x),\dots b_T(x)\right)\right)$ композиция из классификаторов b_t

примеры корректирующих операций

$$F\left(b_1(x),\dots b_T(x)\right)=\frac{1}{T}\sum_{t=1}^{T} b_t(x)$$

простое голосование

$$F\left(b_1(x),\dots b_T(x)\right)=\sum_{t=1}^{T} a_t\cdot b_t(x);\, a_t\in\mathbb{R}$$

взвешенное голосование

композиции классификаторов

$a(x) = C\left(F\left(b_1(x), \ldots b_T(x)\right)\right)$ композиция из классификаторов b_t

примеры корректирующих операций

$$F\left(b_1(x), \ldots b_T(x)\right) = \frac{1}{T}\sum_{i=1}^{T} b_t(x)$$ простое голосование

$$F\left(b_1(x), \ldots b_T(x)\right) = \sum_{i=1}^{T} a_t \cdot b_t(x); a_t \in \mathbb{R}$$ взвешенное голосование

$$F\left(b_1(x), \ldots b_T(x)\right) = \sum_{t=1}^{T} g_t(x) \cdot b_t(x); g_t : X \rightarrow \mathbb{R}$$ смесь алгоритмов

композиции классификаторов

бустинг

$X \subset \mathbb{R}^n$ - объекты ; $y=\{-1;+1\}$ - метки

$b_t(x)$: $X \rightarrow \{-1, 0, +1\}$ классификатор с отказами

взвешенное голосование

$$a(x) = sign\left(\sum_{t=1}^{T} a_t \cdot b_t(x)\right)$$

функционал качества - количество ошибок

$$Q_T = \sum_i \left[\left(y_i \cdot \sum_{t=1}^{T} a_t \cdot b_t(x_i)\right) < 0\right]$$

последовательно добавляем компоненты $a_t b_t(x)$
при этом фиксируем параметры предыдущих компонент

композиции классификаторов

AdaBoost

взвешенное голосование

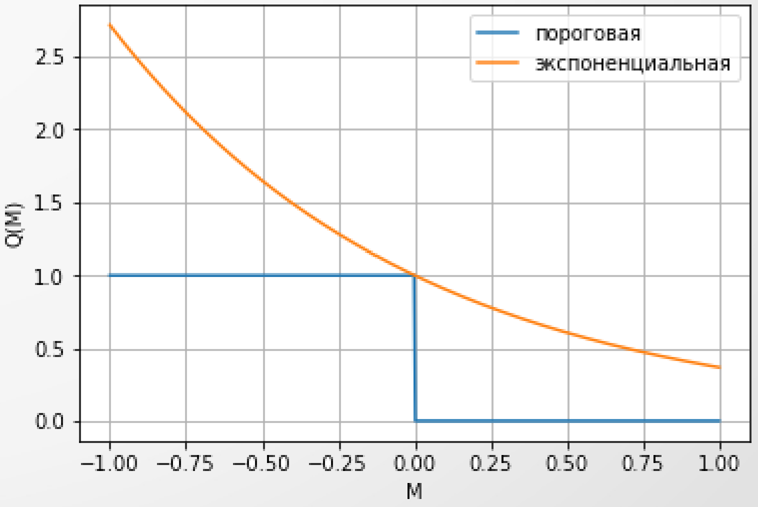
$$a(x) = sign\left(\sum_{t=1}^{T} a_t \cdot b_t(x)\right)$$

функционал качества - количество ошибок

$$Q_T = \sum_i \left[\left(y_i \cdot \sum_{t=1}^{T} a_t \cdot b_t(x_i)\right) < 0\right]$$

функционал качества - пороговая функция
оптимизировать не удобно
заменим его на гладкую аппроксимацию

$$Q_T \leq \widetilde{Q}_T = \sum_i \left[\exp\left(-y_i \cdot \sum_{t=1}^{T} a_t \cdot b_t(x_i)\right)\right]$$



композиции классификаторов

AdaBoost обучение

взвешенное голосование

$$a(x) = sign\left(\sum_{t=1}^{T} a_t \cdot b_t(x)\right)$$

функционал качества

$$Q_T = \sum_i \left[\exp\left(-y_i \cdot \sum_{t=1}^{T} a_t \cdot b_t(x_i)\right)\right]$$

для каждого учебного примера X введём вес w

композиции классификаторов

AdaBoost обучение

взвешенное голосование

$$a(x) = sign\left(\sum_{i=1}^{T} a_t \cdot b_t(x)\right)$$

функционал качества

$$Q_T = \sum_i \left[\exp\left(-y_i \cdot \sum_{i=1}^{T} a_t \cdot b_t(x_i)\right)\right]$$

для каждого учебного примера X введём вес w

сумма весов w примеров x, классифицированных b с ошибкой

$$N(b, W) = \sum_i w_i [b(x_i) = -y_i]$$

сумма весов w примеров x, классифицированных b верно

$$P(b, W) = \sum_i w_i [b(x_i) = y_i]$$

композиции классификаторов

AdaBoost обучение

сумма весов w примеров x, классифицированных b с ошибкой

$$N(b,W)=\sum_i w_i[b(x_i)=-y_i]$$

сумма весов w примеров x, классифицированных b верно

$$P(b,W)=\sum_i w_i[b(x_i)=y_i]$$

функционал качества

$$Q_T=\sum_i\left[\exp\left(-y_i\cdot\sum_{i=1}^{T}a_t\cdot b_t(x_i)\right)\right]$$

Теорема Freund, Schapire (1996)

пусть для вектора весов примеров W существует классификатор b, который классифицирует выборку X лучше чем наугад (P>N) тогда минимум функционала Q_T достигается при следующих параметрах.

$$a_T=\frac{1}{2}\ln\left(\frac{P(b_t,W)}{N(b_t,W)}\right) \qquad b_T=\underset{b}{argmax}\sqrt{P(b,W)}-\sqrt{N(b,W)}$$

Yoav Freund, Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting // Second European Conference on Computational Learning Theory. – 1995.

композиции классификаторов

AdaBoost обучение

$$w_i = \frac{1}{n}$$

начальные значения весов примеров,
n - количество примеров

последовательно обучаем и добавляем компоненты композиции
с учётом весов примеров w

композиции классификаторов

AdaBoost обучение

$$w_i = \frac{1}{n}$$

начальные значения весов примеров,
n - количество примеров

последовательно обучаем и добавляем компоненты композиции
с учётом весов примеров w

вес классификатора
в композиции

$$a_t = \frac{1}{2} \ln\left(\frac{P(b_t, W)}{N(b_t, W)}\right)$$

композиции классификаторов

AdaBoost обучение

$$w_i = \frac{1}{n}$$

начальные значения весов примеров,
n - количество примеров

последовательно обучаем и добавляем компоненты композиции
с учётом весов примеров w

изменение весов примеров
при добавлении классификатора b_t

$$w_i := w_i \cdot \exp\left(-y_i \cdot a_t \cdot b_t(x_i)\right)$$

вес классификатора
в композиции

$$a_t = \frac{1}{2} \ln\left(\frac{P(b_t, W)}{N(b_t, W)}\right)$$

$$w_i := \frac{w_i}{\sum_i w_i}$$

нормируем веса
после коррекции

$$Q_T = \sum_i \left[\underbrace{\exp\left(-y_i \cdot \sum_{i=1}^{T-1} a_t \cdot b_t(x_i)\right)}_{w_i} \cdot \exp\left(-y_i \cdot a_T \cdot b_T(x_i)\right) \right]$$

композиции классификаторов

AdaBoost обучение

$$w_i = \frac{1}{n}$$ начальные значения весов примеров,
n - количество примеров

последовательно обучаем и добавляем компоненты композиции
с учётом весов примеров w

изменение весов примеров
при добавлении классификатора b_t

$$w_i := w_i \cdot \exp\left(-y_i \cdot a_t \cdot b_t(x_i)\right)$$

вес классификатора
в композиции

$$a_t = \frac{1}{2} \ln\left(\frac{P(b_t, W)}{N(b_t, W)}\right)$$

$$w_i := \frac{w_i}{\sum_i w_i}$$ нормируем веса
после коррекции

$$Q_T = \sum_i \left[\underbrace{\exp\left(-y_i \cdot \sum_{i=1}^{T-1} a_t \cdot b_t(x_i)\right)}_{w_i} \cdot \exp\left(-y_i \cdot a_T \cdot b_T(x_i)\right) \right]$$

«трудные» примеры получают больший вес

композиции классификаторов

AdaBoost

композиции классификаторов

примеры базовых классификаторов для AdaBoost

решающие деревья

пороговый классификатор

композиции классификаторов

примеры базовых классификаторов для AdaBoost

решающие деревья

пороговый классификатор

AdaBoost строит длинные композиции из простых классификаторов

для коротких композиции из сложных классификаторов (SVM) результаты AdaBoost хуже

композиции классификаторов

другие методы построения композиций

bagging
обучение по случайным подвыборкам набора примеров,
подвыборки могут пересекаться,
<u>применяем на больших наборах</u>

композиции классификаторов

другие методы построения композиций

bagging
обучение по случайным подвыборкам набора примеров,
подвыборки могут пересекаться,
<u>применяем на больших наборах</u>

rsm (random subspace method)
обучение на случайном подмножестве признаков,
<u>применяем если много признаков</u>

композиции классификаторов

другие методы построения композиций

bagging
обучение по случайным подвыборкам набора примеров,
подвыборки могут пересекаться,
<u>применяем на больших наборах</u>

rsm (random subspace method)
обучение на случайном подмножестве признаков,
<u>применяем если много признаков</u>

bagging и rsm можно комбинировать

композиции классификаторов

схема построения композиции bagging/rsm

выделяем случайное подмножество примеров/признаков

обучаем классификатор

если результат классификатора хороший
то добавляем в композицию

если ошибка композиции уменьшилась
то повторяем
иначе завершение работы

$$a(x) = sign\left(\sum_{i=1}^{T} b_t(x)\right)$$

композиция - простое голосование

$b_t(x)$: $X \to \{-1, 0, +1\}$ классификатор с отказами

композиции классификаторов

медод RandomForest (случайный лес)

bagging над решающими деревьями без pruning (без оптимизации)

признак в каждой вершине выбираем из случайного подмножества размера k всех признаков учебного набора размера n

$k = \sqrt{n}$ для задач классификации

подбираем количество деревьев T по критерию out-of-bag

$$\text{out-of-bag}(a) = \sum_{i=1}^{\ell} \left[\operatorname{sign}\left(\sum_{t=1}^{T} [x_i \notin U_t] \, b_t(x_i) \right) \neq y_i \right] \to \min$$

т.е. проверяем количество ошибок
на учебных наборах других деревьев

композиции классификаторов: литература

git clone *https://github.com/mechanoid5/ml_lectorium.git*

К.В. Воронцов Композиция классификаторов. - курс "Машинное обучение" ШАД Яндекс 2014

Е.С.Борисов Бустинг - композиции классификаторов http://mechanoid.su/ml-adaboost.html

http://www.machinelearning.ru