Ансамбли

Московский физико-технический институт, МФТИ

Москва

Вводные понятия

Примеры ансамблей, усреднение нескольких алгоритмов:

$$a(x) = \frac{1}{N}(b_1(x) + ... + b_N(x)),$$

голосование большинства

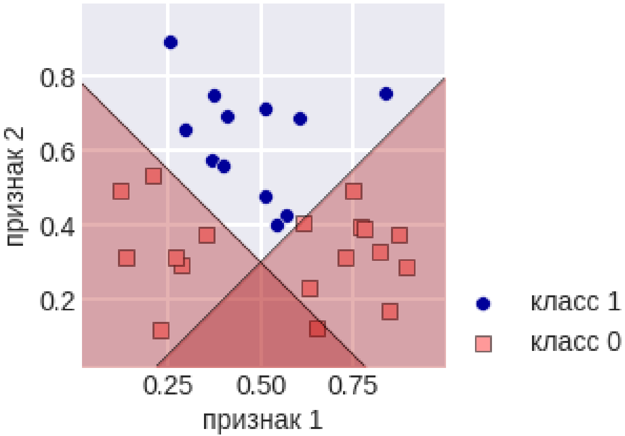
$$a(x) = mode(b_1(x), .., b_N(x)),$$

в общем случае ансамбль записывается как

$$a(x) = b(b_1(x), .., b_N(x)).$$

Одна из целей сделать ансамбль достаточно разнообразным, тогда ошибка отдельных алгоритмов на отдельных обьектах будет компенсироваться корректной работой других алгоритмов.

Расширение возможностей простых алгоритмов



Цель анасамблей: повышения качетсва базовых алгоритмов, повышение разнообразия достигается за счет приемов:

- бэгинг - варьирование обучающей выборки,
- random subspaces - варьирование признаков,
- ECOC - варьирование целевого вектора (деформация целевого признака),
- стекинг - варьирования модели (использование разных моделей)
- случайный лес - варьирование в модели (использование разных алгоритмов в рамках одной)

Обоснования использования ансамблей делятся на

1. статистические
2. вычислительные
3. функциональные

Различные виды ансамблирования 1

Голосование/усреднение	построение независимых алгоритмов и их усреднение (в т.ч при помощи бэгинга) и предварительной деформации ответов
Перекодировка ответов	Специальные кодировки целевых значений и сведение решения задачи к решению решению нескольких задач
Стекинг	построение метапризнаков - ответов базовых алгоритмов на обьектах выборки, обучение на них мета-алгоритмов

Различные виды ансамблирования 2

Бустинг	Построение суммы нескольких алгоритмов. Каждое следующее слагаемое строится с учетом ошибки предыдуших.
"Ручные методы"	Эвристические способы комбинирования ответов базовых алгоритмов.
Однородные ансамбли	Принцип - мета алгоритма - базовые алгоритмы разворачиваются рекурсивно, применяется общая схема оптимизации полученной конструкции. Пример - **нейросети**.

Бутстрап

Пусть дана выборка $X = (x_i, y_i),\ i = 1..K$.
Равномерно возьмем из выборки l обьектов с возррвращением.
Процедуру проведем N раз, сгенерируем N подвыборок X_1,..., X_N. На каждой обучим линейную модель регрессии, получим базовые алгоритмы $b_1(x)$,..., $b_N(x)$.

Пусть известен истинный ответ $y(x)$ и распределение $p(x)$, определим отклонение:

$$\varepsilon_j(x) = b_j(x) - y(x), j = 1, .., N$$

Рассмотрим срднеквадротичное отклонение:

$$\mathbb{E}_x(b_j(x) - y(x))^2 = \mathbb{E}_x\varepsilon_j^2(x).$$

Бутстрап

Рассмотрим композицию алгоритмов

$$a(x) = \frac{1}{N}\sum_{j=1}^{N} b_j(x).$$

Введем предположения

$$\mathbb{E}\varepsilon(x) = 0,$$

$$\mathbb{E}\varepsilon_i(x)\varepsilon_j(x) = 0,\ i \neq j.$$

Бутстрап, среднеквадратичная ошибка

$$E_N = \mathbb{E}_x \left(\frac{1}{N} \sum_{j=1}^{n} b_j(x) - y(x) \right)^2 =$$

$$= \mathbb{E}_x \left(\frac{1}{N} \sum_{j=1}^{n} \varepsilon_j(x)) \right)^2 =$$

$$= \frac{1}{N^2} \mathbb{E}_x \left(\sum_{j=1}^{n} \varepsilon_j^2(x) + \sum_{i \neq j} \varepsilon_i(x) \varepsilon_j(x)) \right)^2 = \frac{1}{N} E_1.$$

Таким образом, усреднение октветов позволило уменьшить средний квадрат ошибок в N раз.

Минимум среднеквадратичного риска

Рассмотрим квадратичную функцию потерь

$$L(y, a) = (y - a(x))^2,$$

соотношение **среднеквадратичного риска**

$$R(a) = \mathbb{E}[(y - a(x))^2] = \int_X \int_Y p(x, y)(y - a(x))^2 dxdy.$$

$$a_*(x) = \mathbb{E}[y|x] = \int_Y yp(y|x)dy = \arg\min R(a)$$

Ошибка метода обучения 1

Определим некторые метод обучения $\mu : (\mathbb{X} \times \mathbb{Y})^l \to \mathbb{A}$. В качестве меры качества метода обучения можно взять усредненный по всем выборкам среднеквадратичный риск алгоритма, выбранного методом μ по выборке:

$$L(\mu) = \mathbb{E}_X[\mathbb{E}_{x,y}[(y - \mu(X)(x))^2]] =$$

$$= \int_{(\mathbb{X}\times\mathbb{Y})^l} \int_{(\mathbb{X}\times\mathbb{Y})} (y-\mu(X)(x))^2 p(x,y) \prod_{i=1}^{l} p(x_i, y_i) dx dy dx_1 dy_1 \ldots dx_l dy_l$$

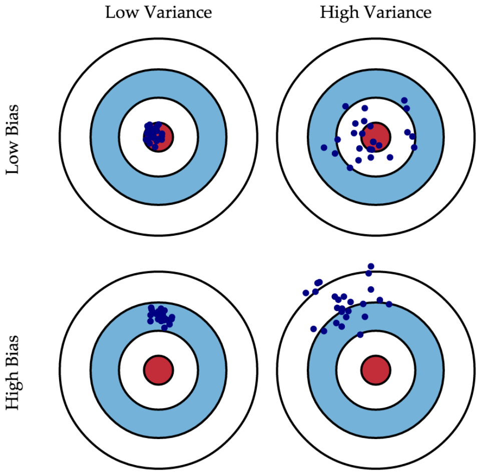
Ошибка метода обучения 2

Распределение обьектов-ответов $p(x, y)$ как правило не известнт, вместо этого определим **метод обучения** $\mu : (\mathbb{X} \times \mathbb{Y})^l \to \mathcal{A}$. Функционал ошибки будет равен

$$L(\mu) = \mathbb{E}_{x,y}[(y - \mathbb{E}[y|x])^2] +$$
$$+ \mathbb{E}_x[(\mathbb{E}_X[\mu(X)] - \mathbb{E}[y|x])^2] + \mathbb{E}_x[\mathbb{E}_X[(\mu(X) - \mathbb{E}_X[\mu(X)])^2]].$$

1. **шум** - ошибка идеального алгоритма,
2. **смещение** - отклонение среднего ответа обученного алгоритма от ответа идеального алгоритма,
3. **дисперсия** - разброс ответов обученных алгоритмов относительно среднего ответа.

Иллюстрация сдвига и разброса



Алгоритм бэгинга

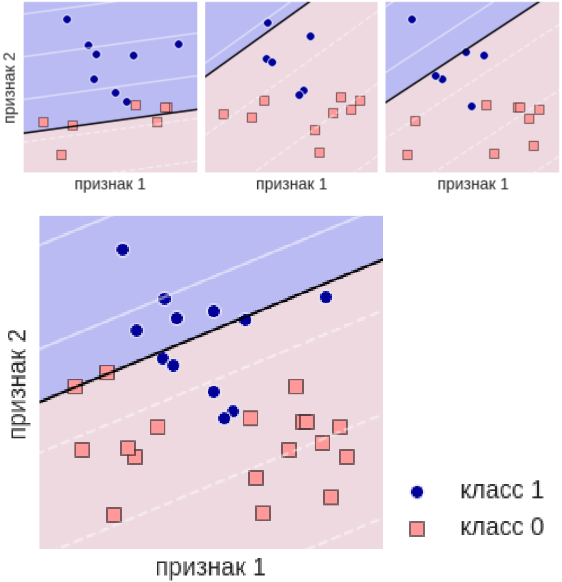
Вероятность быть отобранным при бутсрепе для достаточно большой выборки с возвращением составляет примерно 0.63 (около 36.8

1. Цикл по t - номер базового алгоритма
2. *Взять подвыборку $[X', y']$ обучающей выборки $[X, y]$
3. *обучить tй базовый алгоритм на этой подвыборке $b_t = fit(X', y')$,
4. ансамбль для задач регрессии $a(x) = \frac{1}{n}(b_1(x) + .. + b_n(x))$

Разновидность бэгинга

Бэгинг	подвыборка обучающей выборки берется с помощью бутстепа
Пэстинг	случайная обучающая подвыборка
Случайные подпространства	случайное подмножество признаков
Случайные патчи	одновременно берем случайное подмножество обьектов и празнаков
cross-validation committess	k обучений на (k-1) фолде

Иллюстрация бэгинга



Бэггинг

Бэггинг - bootstrap aggregarion

Пусть задан метод обучения $\mu(X)$, построим метод $\tilde{\mu}(X)$, который генерирует подвыборку $\tilde{X}$: $\tilde{\mu}(X) = \mu(\tilde{X})$, итоговый алгоритм примет вид

$$a_N(x) = \frac{1}{N}\sum_{j=1}^{N} b_j(x) = \frac{1}{N}\sum_{j=1}^{N} \tilde{\mu}(X)(x).$$

Имеем невзвешенное ансамиблирование. Смещение

$$\mathbb{E}_x[(\frac{1}{N}\sum_{j=1}^{N} \mathbb{E}_X[\tilde{\mu}(X)] - \mathbb{E}[y|x])^2] =$$

$$= \mathbb{E}_x[(\mathbb{E}_X[\tilde{\mu}(X)] - \mathbb{E}[y|x])^2].$$

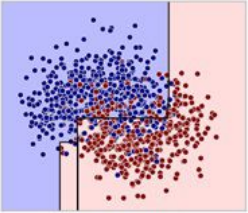
Разброс

$$\mathbb{E}_{x,y}[\mathbb{E}_X[(\frac{1}{N}\sum_{j=1}^{N}\tilde{\mu}(X) - \mathbb{E}_X[\frac{1}{N}\sum_{j=1}^{N}\mu(X)(x)])^2]] =$$

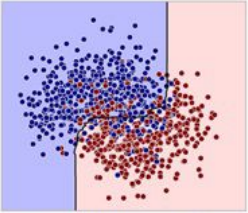
$$= \frac{1}{N}\mathbb{E}_{x,y}[\mathbb{E}_X[(\tilde{\mu}(X) - \mathbb{E}_X[\mu(X)(x)])^2]]+$$

$$+\frac{N(N-1)}{N^2}\mathbb{E}_{x,y}[\mathbb{E}_X[(\tilde{\mu}(X)-\mathbb{E}_X[\mu(X)(x)])\times(\tilde{\mu}(X)-\mathbb{E}_X[\mu(X)(x)])]]$$

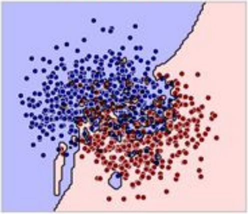
Сравнение бэгинга деревьев и kNN



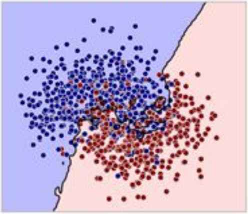
Одно дерево



Бэгинг 100 деревьев



Ближайший сосед



Бэгинг 100 ближайших соседей

Случайные леса

Бэггинг позволяет обьеденить несмещенные, но чувствительные к обучающей выборке алгоритмы в несмещенную композицию с низкой низкой дисперсией.

- n=1,..,N,
- генерируем выборку $\tilde{X}_n$ при помощи бутсрэпа,
- построить решающее дерево $b_n(x)$ по выборке $\tilde{X}_n$,
- * деревья строятся, пока в каждом листе окажется не более n_{min} обьектов,
- * при каждом разбиении выбрирается m начальных признаков из p, оптимальное разделение ищется только среди них,
- возвращается композиция $a(x) = \frac{1}{N}\sum\limits_{j=1}^{N} b_j(x)$.

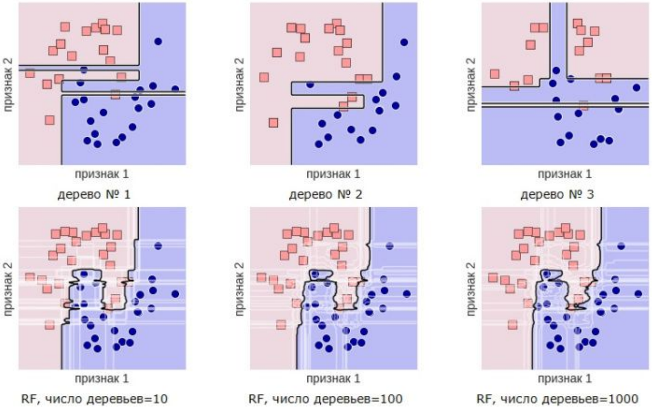
Out-of-bag

При том, что каждое дерево обучается на части обьектов, можно построить фунционал

$$OOB = \sum_{i=1}^{l} L\left(y_i, \frac{1}{\sum_{n=1}^{N}[x_i \notin X_n]} \sum_{n=1}^{N}[x_i \notin X_n] b_n(x_i)\right).$$

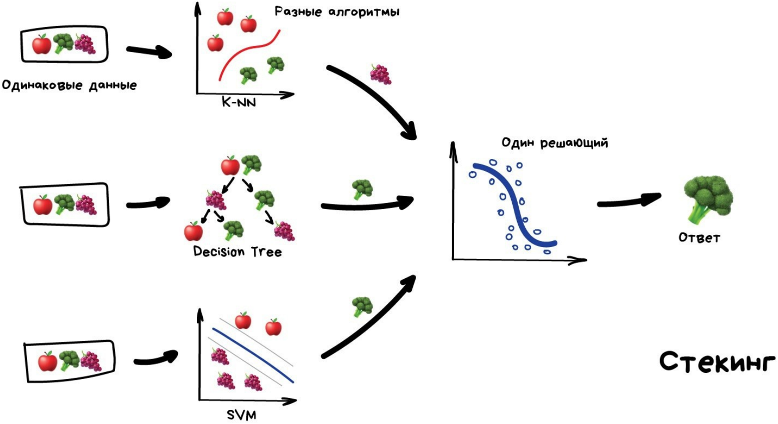
Можно показать, что по мере увеличения числа деревьев, данная оценка стремится к leave-one-out оценке.

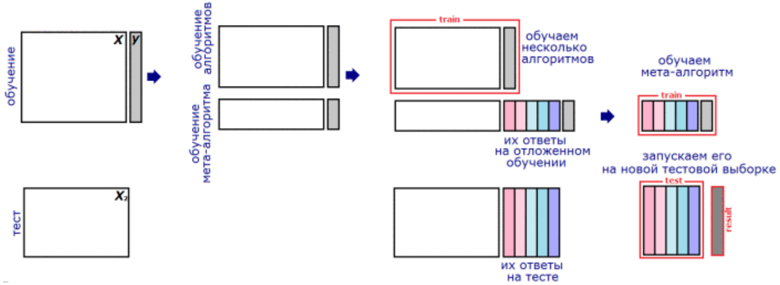
Пример применения решающих деревьев



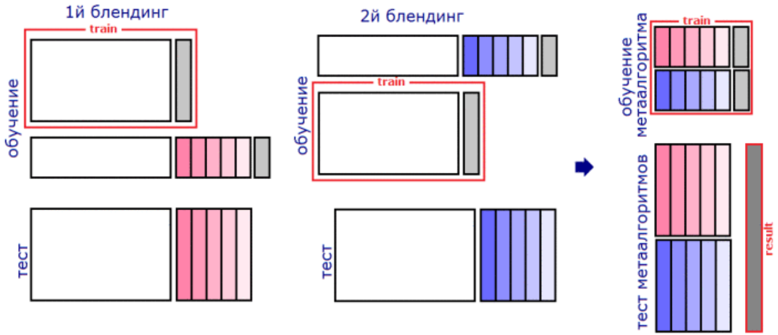
Стекинг

Придуман Д. Волпертом в 1992г.
Блендинг - простая схема стекинга. Обучающая выборка делится на две части. Затем получают их ответы на второй части и на тестовой выборке. Ответы каждого алгоритма можно рассматривать, как **новый признак** (метапризнак). На метапризнаках второй части настраивают метаалгоритм. Затем запускают его на метапризнаках теста и получают ответ.


Разные алгоритмы
Одинаковые данные
K-NN
Один решающий
Decision Tree
Ответ
Стекинг
SVM

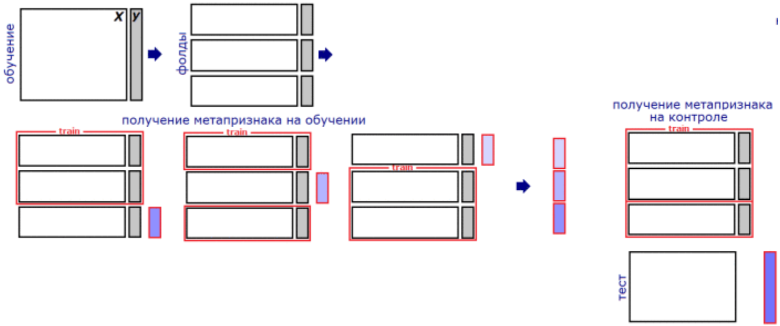


обучение
X
y
тест
X_2
обучение алгоритмов
обучение мета-алгоритма
train
обучаем несколько алгоритмов
их ответы на отложенном обучении
их ответы на тесте
обучаем мета-алгоритм
train
запускаем его на новой тестовой выборке
test
result


1й блендинг
train
обучение
тест
2й блендинг
train
обучение
тест
train
обучение метаалгоритма
тест метаалгоритмов
result

Классический стекинг

Выборку разбивают на части (**фолды**), затем последовательно перебирая фолды обучают базовые алгоритмы на всех фолдах, кроме одного, а на оставшемся получают ответы базовых алгоритмов и трактуют их как значения соответствующих признаков на этом фолде. Для получения метапризнаков обьектов тестовой выборки базовые алгоритмы обучают на всей обучающей выборке и берут их ответы на тестовой.


обучение
X
y
фолды
получение метапризнака на обучении
train
train
train
получение метапризнака
на контроле
train
тест

Бустинг в задаче регрессии

Идея: строить алгоритмы так, чтобы каждая последующая исправляла ошибки предыдушей. Рассмотрим задачу

$$\sum_{i=1}^{l}(a(x_i) - y_i) \to \min_a,$$

как и преждем в итога алгоритмы будут суммироваться:

$$a(x) = \sum_{j=1}^{N} b_j(x),$$

базовый алгоритм

$$b_1(x) = \arg\min_{b\in\mathcal{A}} \sum_{i=1}^{l}(b(x_i) - y_i),$$

далее берем оставшуюся разность

$$s_i^{(1)} = y_i - b_1(x_i),$$

и для нее примененяем указанную процедуру

Заключение

- Композиции позвляют решать сложные задачи, которые плохо решаются отдельными алгоритмами
- буситнг - обучает базовые алгоритмы по очереди,
- бэггинг - обучает базовые алгоритмы независимо,
- стекинг - универсальный подход позволяющий повысить точность алгоритмов