Задание 3

Семен Федотов

Март, 2017

1 Теоретические задачи

Условие 1. *Показать справедливость Bias-variance-noise decomposition*

Доказательство. Вот решение данной задачи. □

Условие 2. *Bagging. Смещение и разброс.*

Доказательство. Ну смещение у нас не изменится, т.к. все алгоритмы в композиции одинаково распределены и, следовательно:

$$E_{X,Y,x,y}(a(x)) = \frac{1}{k}\sum E_{X,Y,x,y}(a_i(x)) = E_{X,Y,x,y}(a_1(x)).$$

Посмотрим на разброс: $Var(a(x)) = \frac{1}{k^2}(\sum \sigma^2 + \sum\limits_{i,j} cov(a_i(x), a_j(x)))$., где $cov_{i,j} = \rho_{i,j} \cdot \sigma^2$, где $\rho_{i,j}$ - коэффициент корреляции между i-тым и j-тым алгоритмом. Отсюда следует, что чем менее они коррелируемы, тем меньше будет разброс. Как раз таки бэггинг и старается сделать менее зависимыми алгоритмы из композиции. □

Условие 3. $\xi_1, \dots \xi_M$ - *о.р.с.в. с* $D\xi_1 = \sigma^2$ *и* $\forall i, j \quad \rho(\xi_i, \xi_j) = \rho > 0$
Доказать, что $D\overline{X} = \rho\sigma^2 + (1-\rho)\frac{\sigma^2}{M}$

Доказательство. Знаем, что дисперсия суммы случайных величин равна сумме их дисперсий и сумме попарных ковариаций. Ковариация и коэффициент корреляции связаны следующим соотношением:

$$\rho = \frac{cov(\xi, \eta)}{\sqrt{D\xi D\eta}}$$

В нашем случае $D\xi = D\eta = \sigma^2 \Rightarrow cov(\xi, \eta) = \rho\sigma^2$.
Посмотрим, что мы получили:

$$D\overline{X} = \frac{1}{M^2}(\sum \sigma^2 + M(M-1)\rho\sigma^2)$$

$$D\overline{X} = \frac{1}{M}(\sigma^2 + (M-1)\rho\sigma^2)$$

$$D\overline{X} = \frac{1}{M}(\sigma^2(1-\rho) + M\rho\sigma^2) = \rho\sigma^2 + \frac{1}{M}\sigma^2(1-\rho) \text{ ЧТД.}$$

□

2 Дополнительные вопросы

Условие 4. *Как сформирован sample_submission.tsv?*

Proof. Просто взято среднее по y в тренировочной выборке.

```
ans = train.y.mean()
```

□

Условие 5. *Посмотреть, как строится DecisionTree[Classifier/Regressor] в sklearn*

Proof. Разбиения в вершинах определяются на основе выбранного информационного критерия + (мы либо выбираем лучший из возможных признаков, либо берем рандомный) + передается минимальное число попавших в лист и тд, все это можно посмотреть в описании этого класса. Пожно предварительно посортить , чтобы искать быстрее разбиения(ведь имеет смысл ходить не по всем точкам, а лишь между/(в точках) выборки признаков), есть также параметр min_impurity_split - который может назначить вершину нодой, если максимум информативности меньше заданной нами границы □

Условие 6. *Эту не успел сделать :(*