Machine Learning and Data Mining

Meta-learning, lecture

Maxim Borisyak

2016/10/09

National Research University
Higher School of Economics

HIGHER SCHOOL OF ECONOMICS
NATIONAL RESEARCH UNIVERSITY

Boosting

Weak learner

Weak learner: a estimator which shows low predictive power on wide range of problems.

General examples:

- cut by a feature;
- Decision Tree with low depth;
- linear SVM;
- logistic regression;
- perceptron.

Set of Weak Learners

Let's focus only on a class of Machine Learning problems, e.g.:

- text classification;
- visual pattern recognition;
- robot control...

In this lecture we consider only the following case:

Set of Weak Learners

Set of weak learners (or weak model) is a set of estimators $\mathcal{M} = \{m_i\}_{i=0}^{N}$, with quality metric $Q(\cdot, \cdot)$ and some $\varepsilon > 0$:

$$\forall \text{probem} \in \text{Problems} : \exists M \subseteq \mathcal{M} : \forall m \in M : Q(m, \text{probem}) > \varepsilon$$

It is preferable that estimators in each subset M is noncorrelated (or weakly correlated).

Boosting

Boosting is a procedure that constructs a strong learner from a set of weak learners.

Common technique:

- construct strong estimator sequentially as an ensemble of weak estimators;
- on each step add new weak estimator;
- new weak estimator is to correct mistakes of current ensemble.

Example

Decision Trees can be viewed as a boosting on cuts by one feature.

Common case:

- For set of weak models $\{e\}$ and loss L:

$$E_n(x) = \sum_{i=1}^{n} \alpha_i e_i(x)$$

$$L(E_1) < L(E_2) < L(E_3) < \cdots < L(E_N)$$

Gradient Boosting

Consider:

- a function F;
- an approximation error (loss) $L_F(\cdot)$;
- set of weak estimators $\mathcal{E} = \{e\}$.

Idea:

- construct
$$E = \sum_{i=1}^{N} \alpha_i e_i$$
which minimizes $L(E)$ better than any estimator $e \in \mathcal{E}$ alone.

Matching pursuit:

- start with base estimator e_0, e.g. $e_0(x) = 0$;

$$E_0 = e_0$$

- for $i = 1 \dots N$:
    - select an estimator e_i and α_i so that:

$$e_i, \alpha_i = \underset{e \in \mathcal{E}, \alpha}{\arg\min}\, L_F(E_{i-1} + \alpha e)$$

    - $E_i = E_{i-1} + \alpha_i e_i$

Matching pursuit with gradient descent:

- start with base estimator e_0, e.g. $e_0(x) = 0$;

$$E_0 = e_0$$

- for $i = 1 \dots N$:
  - select an estimator e_i so that:

$$e_i \approx \frac{\partial L_F}{\partial E_{i-1}}$$

  - select α_i:

$$\alpha_i = \arg\min_{\alpha} L_F(E_{i-1} + \alpha_i e_i)$$

  - $E_i = E_{i-1} + \alpha_i e_i$

- set of training points $D_0 = \{x_j, y_j\}_{j=1}^{M}$;
- loss:

$$L_D(e) = \frac{1}{|D|} \sum_{x,y \in D} l(y, e(x))$$

- weak model $\mathcal{M}$;
- weak learner fitting procedure for some dataset D:

$$\mathrm{fit}(D) = \arg\min_{e \in \mathcal{M}} \mathcal{L}_D(e, D)$$

General Gradient Boosting

- start with base estimator e_0, e.g. $e_0(x) = 0.5$;

$$E_0 = e_0$$

- for $i = 1 \ldots N$:
  - select an estimator e_i so that:

$$e_i = \arg\min_{e \in \mathcal{M}} L_D(E_{i-1} + e_i)$$

  - $E_i = E_{i-1} + e_i$

Gradient Boosting

- approximate gradient $\frac{\partial L_F}{\partial E_{i-1}}$ in training points:

$$\frac{\partial L_F}{\partial E_i} \to D_i = \left\{ \left(x_j, \frac{\partial l_F(y_j, E_i(x_j))}{\partial E_i(x_j)} \right) \right\}_{j=1}^{M}$$

- selecting procedure:

$$e_i \approx \frac{\partial L_F}{\partial E_{i-1}}$$

to fitting:

$$e_i = \mathrm{fit}(D_i)$$

Gradient Boosting

- start with base estimator e_0, e.g. $e_0(x) = 0.5$;

$$E_0 = e_0$$

- for $i = 1 \ldots N$:
  - compute D_i

$$D_i = \left\{ \left(x_j, \frac{\partial l_F(y_j, E_i(x_j))}{\partial E_i(x_j)} \right) \right\}_{j=1}^{M}$$

  - select an estimator e_i so that:

$$e_i = \mathrm{fit}(D_i)$$

  - select α_i:

$$\alpha_i = \arg\min_{\alpha} L_D(E_{i-1} + \alpha e_i)$$

  - $E_i = E_{i-1} + \alpha_i e_i$

$$
\begin{aligned}
L_i &= \sum_j l(y_j, E_i(x_j)) + \Omega(E_i);\\
\Omega(e_i) &= \gamma T_i + \frac{\lambda}{2}\sum_{k=1}^{T_i}(w_k^i)^2.
\end{aligned}
$$

where:

- $\Omega(E_i) = \sum_{j=1}^{N} \Omega(e_j)$ - regularization;
- T_i - number of leafs of i-th tree;
- w_k^i - score of k-th leaf in i-th tree.

Example: Gradient Boosting on Decision Trees

Performing one Newton step (Taylor decomposition up to quadratic term):

$$
L_i \approx \sum_j [
$$

$$
l(y_j, E_{i-1}(x_j)) +
$$

$$
\frac{\partial l(y_j, E_{i-1}(x_j))}{\partial E_{i-1}(x_j)}(E_i(x_j) - E_{i-1}(x_j)) +
$$

$$
\frac{1}{2}\frac{\partial^2 l(y_j, E_{i-1}(x_j))}{\partial E_{i-1}^2(x_j)}(E_i(x_j) - E_{i-1}(x_j))^2
$$

$$
] + \Omega(E_i)
$$

Example: Gradient Boosting on Decision Trees

- $g_j^i = \frac{\partial l(y_j, E_{i-1}(x_j))}{\partial E_{i-1}(x_j)}$
- $h_j^i = \frac{\partial^2 l(y_j, E_{i-1}(x_j))}{\partial E_{i-1}^2(x_j)}$
- $e_i(x_j) = (E_i(x_j) - E_{i-1}(x_j))$
- $L_i \approx L_{i-1}(E_{i-1}) + \Delta L_i(e_i) + \Omega(E_{i-1}) + \Omega(e_i)$

$$
\begin{aligned}
\Delta L_i(e_i) &= \sum_j \left[g_j^i e_i(x_j) + \frac{1}{2} h_j e_i^2(x_j) \right]; \\
\Omega(e_i) &= \gamma T + \frac{\lambda}{2} \sum_{k=1}^{T} w_k^2.
\end{aligned}
$$

Example: Gradient Boosting on Decision Trees

$$
\begin{aligned}
\Delta L_i(e_i) &= \sum_{j=1}^{M} \left[g_j^i e_i(x_j) + \frac{1}{2} h_j e_i^2(x_j) \right] \\
&= \sum_{k=1}^{T} \left[G_k w_k + \frac{1}{2} H_k w_k^2 \right]; \\
\Delta L_i(e_i) + \Omega(e_i) &= \sum_{k} \left[G_k w_k + \frac{1}{2} (H_k + \lambda) w_k^2 \right] + \gamma T;
\end{aligned}
$$

where:

- G_k - sum of g_j within k-th leaf;
- H_k - sum of h_j within k-th leaf.

$$\Delta L_i(e_i) + \Omega(e_i) = \sum_k \left[G_k w_k + \frac{1}{2}(H_k + \lambda) w_k^2 \right] + \gamma T$$

Solution for w_k:

$$w_k^* = -\frac{G_k}{H_k + \lambda}$$

Loss:

$$L^* = -\frac{1}{2} \sum_{k=1}^{T} \frac{G_k^2}{H_k + \lambda} + \gamma T$$

Loss:

$$L^* = -\frac{1}{2}\sum_{k=1}^{T}\frac{G_k^2}{H_k+\lambda} + \gamma T$$

Tree split criteria:

$$\mathrm{gain} = \frac{1}{2}\left\{\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} + \frac{(G_R+G_L)^2}{H_L+H_R+\lambda}\right\} - \gamma$$

where:

- G_L, G_R - G for left and right leafs;
- H_L, H_R - H for left and right leafs.