Compboost: Classes

Daniel Schalk

2018-04-03

The Classes of Compboost

As often pointed out, `compboost` includes many different classes. In this section we want to give a overview about these classes and their functionality. This section first describes the classes that are independent of other classes like the loss class and then goes on to classes which includes other ones like the baselearner factory list class which obviously includes baselearner factories.

To be as flexible as possible, it is important that different classes of the same type (e.g. Bernoulli loss and quadratic loss, both losses) can be called identically. This is required since we are trying to be as generic as possible which results that many classes are child classes of abstract parent classes. Using this polymorphism ensures consistent function calls and pretend a minimal functionlaity of all child classes. This section is also intended to clarify where and how such relationships exists.

For a more detailed documentation which also illustrated the dependencies and between the classes see the `C++` documentation:

Loss Classes

For a theoretical background see section methodology.

`Loss` (Abstract/Parent)

This class defines functions that each child class must have. The child classes contains the concrete implementation. The following functions are defined as virtual, hence these have to be defined within the child classes which derive that virtual functions:

- `arma::vec definedLoss (const arma::vec&, const arma::vec&)` defines the loss function:
$$L(y, f(x))$$
- `arma::vec definedGradient (const arma::vec&, const arma::vec&)` defines the gradient of the loss function with respect to $f(x)$:
$$\frac{\delta}{\delta f(x)} L(y, f(x))$$
- `double constantInitializer (const arma::vec&)` defines how f should be initialized by setting $\hat{f}^{[0]}(x) = c$ as constant. It is possible to choose any constant $c \in \mathbb{R}$ here. The default implementation initialize the algorithm in an loss optimal fashion (if possible):
$$c = \arg\min_{c \in \mathbb{R}} \mathcal{R}_{\text{emp}}(c) = \frac{1}{n} \sum_{i=1}^{n} L(y^{(i)}, c)$$

A special protected member of the `Loss` class is the `custom_offset`. This `double` can be setted by the constructors of the child classes. Calling such a constructor and set a custom offset $c_0 \in \mathbb{R}$ forces the algorithm to initialize the model with that value:
$$\hat{f}^{[0]}(x) = c_0 = c$$

`QuadraticLoss` (Child)

Constructors:

```
QuadraticLoss ();
QuadraticLoss (const double& cusotm_offset);
```

The `QuadraticLoss` class implements the quadratic loss. This loss can be used for regression with $y \in \mathbb{R}$.

Loss Function:

$$L(y, f(x)) = \frac{1}{2} (y - f(x))^2$$

Gradient:

$$\frac{\delta}{\delta f(x)} L(y, f(x)) = f(x) - y$$

Initialization:

$$\hat{f}^{[0]}(x) = \arg\min_{c \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} L\left(y^{(i)}, c\right) = \bar{y}$$

`AbsoluteLoss` (Child)

Constructors:

```
AbsoluteLoss ();
AbsoluteLoss (const double& cusotm_offset);
```

Implements of the absolute loss. This loss can be used for regression with $y \in \mathbb{R}$. It is more robust in terms of outliers.

Loss Function:

$$L(y, f(x)) = |y - f(x)|$$

Gradient:

$$\frac{\delta}{\delta f(x)} L(y, f(x)) = \text{sign}\,(f(x) - y)$$

Initialization:

$$\hat{f}^{[0]}(x) = \arg\min_{c \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} L\left(y^{(i)}, c\right) = \text{median}(y)$$

`BernoulliLoss` (Child)

Constructors:

```
BernoulliLoss ();
BernoulliLoss (const double& cusotm_offset);
```

This loss can be used for binary classification. The coding of y has to be as $y \in \{-1, 1\}$. Note that the algorithm just returns f as real number. To calculate estimator for classes one has to apply the signum function on f. To obtain probabilities another function to map f on $[0, 1]$ is necessary (e. g. the sigmoid function).

Loss Function:

$$L(y, f(x)) = \log\left\{1 + \exp\left(-yf(x)\right)\right\}$$

Gradient:

$$\frac{\delta}{\delta f(x)} L(y, f(x)) = -\frac{y}{1 + \exp\left(yf\right)}$$

Initialization:

$$\hat{f}^{[0]}(x) = \frac{1}{2} \log \left(\frac{p}{1-p} \right)$$

with

$$p = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{y_i > 0\}}$$

`CustomLoss` (Child)

`CustomCppLoss` (Child)

Data Classes

`Data` (Abstract/Parent)

`InMemoryData` (Child)

Baselearner Related Classes

`Baselearner` (Abstract/Parent)

`PolynomialBlearner` (Child)

`PSplineBlearner` (Child)

`CustomBlearner` (Child)

`CustomCppBlearner` (Child)

`BaselearnerTrack`

`BaselearnerFactory` (Abstract/Parent)

`PolynomialBlearnerFactory` (Child)

`PSplineBlearnerFactory` (Child)

`CustomBlearnerFactory` (Child)

`CustomCppBlearnerFactory` (Child)

`BaselearnerFactoryList`

Logger Related Classes

`Logger` (Abstract/Parent)

`IterationLogger` (Child)

`TimeLogger` (Child)

`InbagRiskLogger` (Child)

`OobRiskLogger` (Child)

`Loggerlist`

Optimizer Classes

`Optimizer` (Abstract/Parent)

`GreedyOptimizer` (Child)

Compboost Class