Method of Ensembles in Machine Learning

Dr. Kalidas Y., IIT Tirupati

By the end of this lecture you would have understood the concept of ensembles, and three popular paradigms – bagging, boosting and random forests.

130) key phrase… “Bagging”

- Bootstrap Aggregating
- Subsets of Data are used to build Several Classifiers (subsets of rows)
- Aim: Reduce Variance
- Methodology:
  - Build several models
  - Regression:
    - Take Average
  - Classification:
    - Voting

ALGORITHM 2 Bagging methodology.

Require: X /*Data set*/

$\Gamma = []$

for $i = 1 : M$ **do**

$X_{sub} \leftarrow \textbf{sample}(X)$

$\Gamma \leftarrow \Gamma \odot \textbf{train}(X_{sub})$

end for

if classification problem **then**

$y \leftarrow \textbf{voting}(\{\gamma(x) : \forall \gamma \in \Gamma\})$

else

$y \leftarrow \textbf{mean}(\{\gamma(x) : \forall \gamma \in \Gamma\})$

end if

where incrementally, over steps, several weak classifiers are combined so a

131) key phrase… "Boosting"

- Combination of Weak Classifiers to form a Strong Classifier
- Weak classifier: > 50% accuracy (it can be 50.0001 even, but definitely greater)
  - Example – *Audience Poll* in *Kaun Banega Crorepathi*
  - How different is it from Voting used in Bagging?
  - *Bagging: Does not have the > 50% requirement!*
- *Boosting requires each classifier > 50% accuracy*
- *Boosting requires each regressor > 50% score (whatever be the 0 to 1 metric)*
- Aim: Increase Accuracy
- Methodology:
  - Classification and Regression
  - Notion of weights comes into picture

ALGORITHM 3 Boosting methodology.

Require: X /*Data set*/

$\Gamma_0 = 0$ /*List of classifiers so far*/

$Error = L(\Gamma(X), y)$ /*Loss function*/

for $i = 1 : M$ **do**

$\Gamma_i \leftarrow \Gamma_{i-1} + \alpha \times f(x)$ /*Add a new learner*/

$f_i, \alpha_i \leftarrow \arg\min_{f,\alpha} L(\Gamma_i(X), y)$ /*Reduce loss further*/

/*For a convex loss function*/

Setting $\dfrac{\partial L(\Gamma_i(X), y)}{\partial f} = 0 \rightarrow f_i$

Setting $\dfrac{\partial L(\Gamma_i(X), y)}{\partial \alpha} = 0 \rightarrow \alpha_i$

end for

$y = \Gamma_M(x)$ /*to predict for new input vector*/

ALGORITHM 5 AdaBoost algorithm.

$W_0(i) = 1/N(\forall i \in [1 \ldots N])$
for $m = 1 : M$ **do**
  $h_m = \arg\min_h \sum_{i=1}^{i=N} W_{m-1}(i) * 1\{y_i \neq h(x_i)\}$
  $\epsilon_m = \sum_{i=1}^{i=N} W_{m-1}(i) * 1\{y_i \neq h_m(x_i)\}$
  $\alpha_m = 1/2 * \log\left(\frac{1-\epsilon_m}{\epsilon_m}\right)$
  for $i = 1 : N$ **do**
    if $y_i = h_m(x_i)$ **then**
      $\gamma = \frac{1}{1-\epsilon_m}$
    else
      $\gamma = \frac{1}{\epsilon_m}$
    end if
    $W_m(i) = 1/2 * W_{m-1}(i) * \gamma$
  end for
end for
$y = \sum_{m=1}^{m=M} \alpha_m * h_m(x)$

132) key phrase…
“Ada Boost Algorithm”

upati

Edit View Window Help

Open 108 (127 of 498) 125%

between predicted and actual outputs. Let $L(F(x), y) = 1/2 * (y - F(x))^2$ be the squared loss function. Then the following derivations compute new values of F in Eq. (40).

133) key phrase…
“Gradient Boosting Algorithm”

$$L(F(x),y) = 1/2 * (y - F(x))^2$$

$$\frac{\partial L(F(x),y)}{\partial F(x)} = 1/2 * 2 * (y - F(x)) * -1$$

$$\Rightarrow \nabla L(F(x),y) = \frac{\partial L(F(x),y)}{\partial F(x)} = (F(x) - y) \Rightarrow F^{new}(x)$$

$$= F^{old}(x) - \nabla L(F(x),y)|_{F(x)=F^{old}(x)}$$

$$F^{new}(x) = F^{old}(x) - predicted(F^{old}(x) - y) \tag{40}$$

Building sequence of classifiers: The steps in building sequence of classifiers is as follows.

- Let $F_1(x)$ be the first classifier built over the data set
- Let $e_1(x) = F_1(x) - y$ be the classifier or regressor for the error
- Let $F_2(x) = F_1(x) - e_1(x)$ be the updated classifier
- Let $e_2(x) = F_2(x) - y$ be the classifier or regressor for the error on the update classifier
- Let $F_3(x) = F_2(x) - e_2(x)$ be the updated classifier
- and so on
- Let $F_{M+1}(x) = F_M(x) - e_M(x)$
- Then we can expand $F_{M+1}(x) = F_1(x) - \sum_{i=1}^{i=M} e_i(x)$

For any other loss function of the form $L(F(x) - y)$ if the function is not a constant function, then the gradient term, $\nabla L(F(x) - y) \propto (F(x) - y)$. There

ALGORITHM 4 Random forest methodology.

Require: X /*Data set*/

$\Gamma_0 = []$ /*List of classifiers so far*/

$\chi = cols(X)$ /*set of columns*/

for $i = 1 : M$ **do**

$\chi_i = subset(\chi)$ /*random subset*/

$h^* = L(h(X[\chi_i]), y)$ /*best classifier on subset of columns*/

$\Gamma_i \leftarrow \Gamma_{i-1} \odot h^*$

end for

if classification problem **then**

$y \leftarrow \textbf{voting}(\{\gamma(x) : \forall \gamma \in \Gamma\})$

else

$y \leftarrow \textbf{mean}(\{\gamma(x) : \forall \gamma \in \Gamma\})$

end if

134) key phrase…

“Random Forest Algorithm”

TABLE 4 Comparison of ensembling methods.

Aspect	Bagging	Boosting	Random forest
Training data for building constituent classifiers	Multiple subsets	Whole data	Whole data
Typical complexity of the constituent classifiers	Complex	Weak	Complex
Variance reduction	Yes	No	Yes
Bias reduction	No	Yes	No
Complexity control	Constituent classifiers	Ensemble size	Constituent classifiers and ensemble size