Ensemble Methods II

AW

Lecture Overview

1. Boosting
2. Bagging
3. Random Forests
4. Diversity
5. Ensemble Pruning

Boosting: Exponential Loss

- Used in AdaBoost algorithm to update distributions

Ground-truth function (real classification)

Classifiers h1, h2...

$$\ell_{\exp}(h \mid \mathcal{D}) = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}[e^{-f(\boldsymbol{x})h(\boldsymbol{x})}]$$

- Used because of its simplicity in partial derivations and its consistency with the goal to minimize classification error
- For two classes -1 and +1 we can get simple update formula:

$$H(\boldsymbol{x}) = \frac{1}{2} \ln \frac{P(f(x) = 1 \mid \boldsymbol{x})}{P(f(x) = -1 \mid \boldsymbol{x})}$$

Boosting: Exponential Loss

$$H(\boldsymbol{x}) = \frac{1}{2} \ln \frac{P(f(x) = 1 \mid \boldsymbol{x})}{P(f(x) = -1 \mid \boldsymbol{x})}$$

which means that we can easily choose class -1 if numerator is greater than denominator and 1 otherwise

$$\begin{aligned}
\texttt{sign}(H(\boldsymbol{x})) &= \texttt{sign}\left(\frac{1}{2} \ln \frac{P(f(x) = 1 \mid \boldsymbol{x})}{P(f(x) = -1 \mid \boldsymbol{x})}\right) \\
&= \begin{cases} 1, & P(f(x) = 1 \mid \boldsymbol{x}) > P(f(x) = -1 \mid \boldsymbol{x}) \\ -1, & P(f(x) = 1 \mid \boldsymbol{x}) < P(f(x) = -1 \mid \boldsymbol{x}) \end{cases} \\
&= \underset{y \in \{-1,1\}}{\arg\max} P(f(x) = y \mid \boldsymbol{x}),
\end{aligned}$$

Lecture Overview

1. Boosting
2. Bagging
3. Random Forests
4. Diversity
5. Ensemble Pruning

Bagging: Soft Voting

- Used for classifiers that produce class probability outputs

$$(h_i^1(\boldsymbol{x}), \ldots, h_i^l(\boldsymbol{x}))^\top$$

l-dim. vector where each value is estimate of the probability of x belonging to class l

- Each classifier's vote is vector consisting of probabilities for each class
- Simple soft voting then generated using the combined output by averaging all individual outputs
- **Weighted soft voting** can be also considered in different variants explained in the following slides

$$H^j(\boldsymbol{x}) = \frac{1}{T} \sum_{i=1}^{T} h_i^j(\boldsymbol{x})$$

Bagging: Weighted Soft Voting

- Method 1:

$$H^j(\boldsymbol{x}) = \sum_{i=1}^{T} w_i h_i^j(\boldsymbol{x})$$

Each classifier is weighted according to its performance

- Method 2:

$$H^j(\boldsymbol{x}) = \sum_{i=1}^{T} w_i^j h_i^j(\boldsymbol{x})$$

- Each classifier has j number of weights where j is amount of classes used.
- Hence, each class for each classifier has different importance
- There is also another method where weight is assigned to each example of each class for each classifier, but it is not used in practice due to amount of coefficients that need to be computed.

R - Bagging

- Bagged CART:

```
control <-
trainControl(method="repeatedcv",
number=10, repeats=3)
fit.treebag <- train(Class~.,
data=dataset, method="treebag",
metric="Accuracy",
trControl=control)
```

Uses repeated 10 fold cross-validation with three repetitions

Classification accuracy

Also possible to put control line directly here

Random Forest

- Relatively new ensemble method designed for decision tree classifiers like rpart
- Extension of Bagging with randomized feature selection
- Algorithm similar to Bagging with a change.
- First, we generate multiple decision trees
- Each tree is created using random vectors that are generated from a fixed probability distribution
- Then, we combine predictions of each tree => random forest

Lecture Overview

1. Boosting
2. Bagging
3. Random Forests
4. Diversity
5. Ensemble Pruning

R - Random Forest

- ```
  control <-
  trainControl(method="repeatedcv",
  number=10, repeats=3)


  fit.randomForest <- train(Class~.,
  data=dataset, method="rf",
  metric="Accuracy",
  trControl=control)
  ```
- The only difference from the CART is an **rf** value in **method** parameter
  ```

R – Random Forest cont.

- Example using Iris data from class:

```
library(sets)
library(caret)
library(randomForest)

set.seed(365) # good to set seed so each run yields the same results

x<-sample(1:150, 50, replace = T) #randomly select record #'s with
replacement
y<-as.integer(as.set(1:150)-as.set(x)) # take a complement
irisTL<-iris[x,]; # learning
irisTC<-iris[y,] # testing

# Random Forest
randomForest <- train(Species ~ ., data = irisTL,
        control = trainControl(method="repeatedcv", number=10,
repeats=3))

# prediction
pred.randomForest <- predict(randomForest,newdata=irisTC)
acc.randomForest <-
sum(pred.randomForest==irisTC$Species)/dim(irisTC)[1]
acc.randomForest
```

- Accuracy is 95%

Lecture Overview

1. Boosting
2. Bagging
3. Random Forests
4. Diversity
5. Ensemble Pruning

Diversity

- No point in combining classifiers that are very similar
- Hence, there is a need for evaluating diversity of each classifier.
- Not an easy task because we are evaluating classifiers trained on the same dataset, so their results are usually correlated

Diversity Measures

- **Pairwise Measures**
- Measures pairwise similarity/dissimilarity between two classifiers
- One variant is **Disagreement Measure** by Skalak and Ho:

	$h_i = +1$	$h_i = -1$
$h_j = +1$	a	c
$h_j = -1$	b	d

Contingency table for classifiers hj and hi

$$dis_{ij} = \frac{b+c}{m}$$

Greater values of dis means greater diversity

Diversity Measures

- **Non-Pairwise Measures**
- These measures try to assess diversity directly.
- **Kohavi-Wolpert Variance**
- Originates from the bias-variance decomposition of the error of the classifier.

$$var_{\boldsymbol{x}} = \frac{1}{2}\left(1 - \sum_{y \in \{-1,+1\}} P(y \mid \boldsymbol{x})^2\right)$$

Variability of the predicted class **y** on an instance **x**

Lecture Overview

1. Boosting
2. Bagging
3. Random Forests
4. Diversity
5. Ensemble Pruning

Ensemble Pruning

- Not necessary to combine all classifiers
- Choose only a subset of them that will create the "ensemble"
- Used only on parallel methods such as Bagging
- Intractable to prune Boosting methods (Tamon and Xiang, 2000)

Ensemble Pruning - Methods

- Ordering-based pruning
  - Order classifiers based on defined criteria and only select few classifiers would be chosen for the ensemble.
- Clustering-based pruning
  - Partition classifiers into groups based on their similarity and choose one from each group.
- Optimization-based pruning
  - Find a subset of classifiers that minimizes or maximizes given objective. One possibility is to use heuristics for the objective.

Ensemble Pruning

- **Advantages**
  - Smaller ensembles
  - Faster computing
  - Improved performance

Reading

- L. Rokach. Ensemble Methods for classifiers – chapter in Data Mining and Knowledge Discovery Handbook https://datajobs.com/data-science-repo/Ensemble-Methods-[Lior-Rokach].pdf
- TSK textbook, section 5.6
- R implementation: http://machinelearningmastery.com/machine-learning-ensembles-with-r/
- R caret package documentation: https://cran.r-project.org/web/packages/caret/vignettes/caret.pdf