ML Methods for KW Weights

– Methods overview –

Christoph Kern

November 27, 2018

1 Classification and Regression Trees (CART)

Breiman et al. (1984)

- Building blocks (base learner) for random forests, boosting trees
- Basic idea: Recursive binary splitting – repeatedly split the predictor space into subgroups that are homogeneous with respect to the outcome
- Splitting criterion for classification trees: Reduction in impurity

$$\Delta I_{Gini}(s, \tau) = I_{Gini}(\tau) - p(\tau_L) I_{Gini}(\tau_L) - p(\tau_R) I_{Gini}(\tau_R)$$

with

$$I_{Gini}(\tau) = \sum_{k=1}^{K} \hat{p}_{k_\tau}(1 - \hat{p}_{k_\tau})$$

- Final tree is a set of regions (terminal nodes) with associated scores for prediction

$$\mathcal{T}(x; \Theta) = \sum_{m=1}^{M} \gamma_m I(x \in \tau_m)$$

Algorithm 1: Tree growing process

Parameter : Stopping criteria
Initialization : Assign training data to root node

```
if stopping criterion is reached then
    end splitting;
else
    find the optimal split point;
    split node into two subnodes at this split point;
    for each node of the current tree do
        continue tree growing process;
    end
end
```

2 Model-based Recursive Partitioning (MOB)

Zeileis et al. (2008)

- Combines parametric regression with tree growing
- Utilizes parameter instability tests to determine whether distinct models for different subgroups are needed
- Final model structure: A tree with GLMs in the terminal nodes
- Tuning parameters
  - p-value threshold (`alpha`)
  - Minimum number of observations in a node (`minsplit`)
  - Maximum depth of the tree (`maxdepth`)

Algorithm 2: Recursive partitioning with GLMs

Parameter : p-value threshold
Initialization : Fit initial model using all observations

```
Perform M-fluctuation tests for each partitioning variable;
if minimum p-value exceeds threshold then
    end partitioning;
else
    choose partitioning variable associated with the smallest p-value;
    find the optimal split point;
    split node into two subnodes at this split point;
    for each node of the current tree do
        continue partitioning process;
    end
end
```

3 Random Forests (RF)

Breiman (2001)

- Combines many CART-like trees into an ensemble
- Uses bootstrapping and sampling of predictors to build individual trees
- Final model: A large number of trees whose predictions are averaged over trees
- Tuning parameters
  - Number of predictors to sample at each split point (`mtry`)
  - Number of trees (`num.trees`)
  - Minimum number of observations in a node (`min.node.size`)

Algorithm 3: Grow a Random Forest

```
Parameter : Number of trees B, predictor subset size m, stopping criteria
for b = 1 to B do
    draw a bootstrap sample from the training data;
    assign sampled data to root node;
    if stopping criterion is reached then
        end splitting;
    else
        draw a random sample m from the p predictors;
        find the optimal split point among m;
        split node into two subnodes at this split point;
        for each node of the current tree do
            continue tree growing process;
        end
    end
end
```

4 Extremely Randomized Trees (XTREE)

Geurts et al. (2006)

- Alternative approach to grow a tree ensemble
- Unlike RF, uses whole sample to build each tree but randomly samples split points for each sampled predictor
- Final model structure: Same as RF
- Tuning parameters
  - Number of predictors to sample at each split point (`mtry`)
  - Number of random splits to consider for each sampled predictor (`num.random.splits`)
  - Number of trees (`num.trees`)
  - Minimum number of observations in a node (`min.node.size`)

5 Boosting Trees (GBM)

Friedman et al. (2000), Friedman (2001)

- Class of ensemble methods that focuses on sequential learning
- Trees are grown in sequence using the pseudo-residuals given the previous trees as the outcome

- Final model structure: A sequence of trees whose predictions are added up over trees
- Tuning parameters
  - Number of trees (`n.trees`)
  - Maximum depth of each tree (`interaction.depth`)
  - Shrinkage applied to terminal node estimates of each tree (`shrinkage`)
  - Fraction of randomly sampled observations to grow next tree (`bag.fraction`)

Algorithm 4: Gradient Boosting for regression

Parameter : Number of trees T, interaction depth D, shrinkage λ
Initialization : Use $\bar{y}$ as initial predicted values

for $t = 1$ *to* T **do**
  compute residuals based on current predictions;
  assign data to root node, using the residuals as the outcome;
  while *current tree depth* $< D$ **do**
    tree growing process;
  end
  compute the predicted values of the current tree;
  add the (λ-)shrunken new predictions to the previous predicted values;
end

6 More methods...

- Model-based Boosting (Hofner et al. 2014)
- Bayesian Additive Regression Trees (Chipman et al. 2010)
- Support Vector Machines (Cortes and Vapnik 1995)
- ...

References

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Monterey, CA: Brooks/Cole Publishing.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2):337–407.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1):3–42.

Hofner, B., Mayr, A., Robinzonov, N., and Schmid, M. (2014). Model-based boosting in R: a hands-on tutorial using the R package mboost. *Computational Statistics*, 29(1-2):3–35.

Zeileis, A., Hothorn, T., and Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17(2):492–514.