

CSCI 5622 Machine Learning

ML Ensembles; Random Forests

DATE	READ	DUE
Today, Oct 14	Random Forests	Literature Review
Mon, Oct 19	Hierarchical Clustering	Peer Fdbk Lit Rev
Wed, Oct 21	<i>K</i> -means	Exp. 1 Progress Rep

www.RodneyNielsen.com/teaching/CSCI5622-F09/

Instructor: Rodney Nielsen

Assistant Professor Adjunct, CU Dept. of Computer Science

Research Assistant Professor, DU, Dept. of Electrical & Computer Engr.

Research Scientist, Boulder Language Technologies

ML Ensemble Learning

- Train multiple classifiers
- Use them all in some combination to make the final predictions

- Improve performance by aggregating decisions from classifiers trained on different sets of data
- Generally have a single training set
- Train on several random bootstrap samples

ML AdaBoost Algorithm

- $D_1(i) \leftarrow 1/N$
- For $t = 1..T$
 - $h_t \leftarrow \text{WeakLearn}(D_t)$
 - $\varepsilon_t \leftarrow \sum_{i:h_t(x_i) \neq y_i} D_t(i)$
 - $\alpha_t \leftarrow 0.5 \ln((1-\varepsilon_t)/\varepsilon_t)$
 - $D_{t+1}(i) \leftarrow D_t(i) \exp(-\alpha_t y_i h_t(x_i)) / Z_t$
- $h(x) \leftarrow \text{sign}(\sum_{t=1..T} \alpha_t h_t(x_i))$

Random Forests

- A random forest is a classifier consisting of a collection of tree-structured classifiers $\{h(\mathbf{x}, \Theta_k), k = 1, \dots\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input \mathbf{x} .
- Random training set, Bagging (Breiman, 1996)
- Random split selection (Dietterich, 1998)
- Random output (Breiman, 1997)
- Random feature selection for trees (Ho, 1998)
- Random feature selection for nodes (Amit & Geman, 1997)

- Claim overfitting is not a problem
 - But in practice, this can still be a problem
- Claim accuracy is insensitive to the number of random features
 - But in practice, this can still be a problem
- Performance depends on two features:
 - Strength of the tree
 - Dependence between trees
- Out-of-bag estimation

ML RF Generalization Error

- Strength of forest: $strength = E_{\mathbf{X}, \mathbf{Y}} margin(\mathbf{X}, \mathbf{Y})$
- $margin(\mathbf{X}, \mathbf{Y}) = P_{\Theta}(h(\mathbf{X}, \Theta) = \mathbf{Y}) - \max_{j \neq \mathbf{Y}} P_{\Theta}(h(\mathbf{X}, \Theta) = j)$
- $PE^* = var(margin)/s^2, s = strength$
- $PE^* \leq mean_correlation (1 - s^2)/s^2$
- The two factors in the generalization error are:
 - The strength of the classifiers in the forest
 - The correlation of their raw margins

ML Random Forest Performance

- $err_{RandomSplitSelection} < err_{Bagging}$
- $err_{RandomNoise} < err_{Bagging}$
- $err_{RandomSplitSelection} < err_{Bagging}$
- $err_{Boosting} < err_{RSS, RN, or Bagging}$
- $err_{Bagging+RFtrSel (AKA RFs)} < err_{Boosting}$

- To improve performance randomness must:
 - Have low correlation
 - Maintain strength

ML Random Forest Properties

- $err_{Breiman's\ Random\ Forests} < err_{Boosting}$
- Can be robust to noise and outliers
 - But this is not always the case
- Faster than Bagging or Boosting
- Provides unbiased error estimates without a validation set and variable importance
- Simple and easily parallelized

ML Random Forests: An/The Algorithm

- Bagging
- Random feature subset evaluation at each node in the tree
- No pruning

ML RFs: Why Bagging

- Claim that it improves accuracy of the tree
 - But not in some large scale experiments
- Provide an estimate of the generalization error
 - Using out-of-bag instances
- Allows calculation of strength and correlation
 - This is only meaningful if these are used in some way, which they usually are not

ML Out-of-Bag Error Estimation

- Training dataset = T
- Bootstrap training datasets = $\{T_k\}$
- Learned classifiers = $\{h(\mathbf{x}, T_k)\}$

ML Out-of-Bag Error Estimation

- Out-of-bag (OOB) instance for T_k
 - An instance in T that was not used to create T_k
- OOB error estimate
 - Average error rate on out-of-bag instances over $\{T_k\}$
- Breiman (1996), empirically shows that error estimates based on OOB data are as accurate as those based on a held-out set of size $|T|=N$
 - Eliminates need for a test set?

ML OOB Error Estimation

- ~37% of T not included in any given T_k
- Claim unbiased error estimate

ML Forreest-RI Experiments

- Forreest-RI
 - Number of random features = 1 or $\text{floor}(1 + \log_2 d)$
 - Used OOB error estimate to select between above
- Used 20 datasets:
 - 13 small UCI, 3 large, 4 synthetic
- RF outperformed AdaBoost on 12/20 datasets
- Selecting just one feature was almost as good as selecting the best out of $\text{floor}(1 + \log_2 d)$
- Extremely fast

ML Forrest-RC Experiments

- Forrest-RC
 - Create a linear combination of 3 features:
 - Give each a random coefficient in $[-1,+1]$ and use the sum
 - Evaluate 2 and 8 of these combinations
 - Used OOB error estimate to select between 2 or 8
- Same 20 datasets
- RF outperformed AdaBoost 14w-3L-3t
- Evaluating 2 combinations was as good as 8 on all but the three large datasets

ML Categorical Attributes in RFs

- Select a random subset of possible values and create a binary attribute that is 1 if the value is in that subset and 0 otherwise
- Increase probability of selection to $(|V|-1)$

ML Emp Eval of Strength & Correlation

- Forest-RF on sonar data (60 inputs, 208 examples)
- Using from 1 to 50 inputs
- In each of 80 iterations, 10% data was held out to test
- F , # of random attrs eval, was varied from 1 to 50
- For each value of F , 100 trees were grown to form a RF
- Recorded test set error, strength, correlation, etc.
- Averaged results over the 80 repetitions
- Altogether, 400k trees were grown in this experiment

ML RFs and Noise

- Dietterich (1998) demonstrated that random changes to the training labels has much more impact on Boosting than on Bagging or Random Split Selection
- Forest-RI and Forest-RC also performed much better than AdaBoost when 5% of the labels were changed in 8/9 datasets

ML Project Discussion

- Questions?
- Office hours: will stay until all questions are addressed
- Or make an appointment for Friday