Feature-Budgeted Random Forest

Feng Nan, Joseph Wang, Venkatesh Saligrama

Center for Information and Systems Engineering
Boston University
Supervised Learning

- Training Data: \((x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\)
  - \(X\): Features, \(Y\): Labels (ex: \(\{0, 1\}\))

- Learn \(f: X \rightarrow Y\)
  - Goal: Min Risk (complexity constrained)

\[
  f_{\ast} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(y^{(i)}, f(x^{(i)}))
\]

- Test time: \(f(x)\)
  - Entire feature vector used
Feature Costs

- \( X \): vector of features
- Features have different Costs
  - Computation, Acquisition
- Goal: Use cheap features if possible
Computational Costs

- $\mathbf{x} = [x_1, x_2, \ldots, x_K]$, $x_k$ - may be high dimensional

- **feature costs:**
  - $c_1, c_2, \ldots, c_K$
  - time, computing resources, money, ...
  - expensive features = more informative
  - cheap features = less informative
Sensor (Feature) Acquisition Costs

- Standoff images of subjects (people) wearing explosive devices underneath clothing
- **Classification objective:** is subject concealing a threat?
- Sensor with different costs
- Each Sensor Produces high-dim. feature

**Goal:**
- Selectively use features
- Minimize Avg. Error & Avg. Feature Cost
Supervised Learning Under Test Time Budget

\[
\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(y^{(i)}, f(x^{(i)})), \quad \text{s.t.} \quad \frac{1}{n} \sum_{i=1}^{n} C(f, x^{(i)}) \leq B.
\]

- **Family of classifiers**
- **Loss function**
- **Cost for using classifier f on x**
- **Total budget**

\[C(f,x): \text{sum of costs of each feature used by f on x}\]
Previous Approaches

**Cascade Model** [Viola&Jones, 2001; Zhang&Zhang, 2010; Chen et al., 2012; Trapeznikov & Saligrama, 2013]
- fixed feature order: cheap to expensive
- decision functions at each node and classifiers at each leaf

**Tree Model**
- fixed tree structure [Xu et al., 2013; Kusner et al., 2014]
- fixed tree structure & feature order [Wang et al., 2014b;a]
- decision functions at each node and classifiers at each leaf
- convex relaxations for joint optimizations

New Features

classify

Stop
Our Approach

- Learn many **decision trees** using random data subsets
- Majority vote for final prediction
Costs On Decision Tree

- each feature has a cost
- $C(f,x)$: sum of feature costs from root to leaf

$C(f,x1) = \text{cost(Avg. pixel intensity)} + \text{cost(Edge-based feature)}$

$C(f,x2) = \text{cost(Avg. pixel intensity)} + \text{cost(SIFT)} + \text{cost(CNN)}$

$$\min_{f \in F} \frac{1}{n} \sum_{i=1}^{n} L(y^{(i)}, f(x^{(i)})), \text{ s.t. } \frac{1}{n} \sum_{i=1}^{n} C(f, x^{(i)}) \leq B.$$
Costs On Random Forest

- Decision tree tends to overfit – use random forest

\[ C(f, x) : \text{sum of costs of unique features used by all trees on } x \]

\[
\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(y^{(i)}, f(x^{(i)})), \quad \text{s.t.} \quad \frac{1}{n} \sum_{i=1}^{n} C(f, x^{(i)}) \leq B.
\]
Key:
Need a new method to build each tree with small cost.
- Account for feature cost
- Theoretical guarantees
- Empirical performance
Growing Low-Cost Trees

- Randomly sample training data

- Split data recursively into subsets
  - For each feature \( t \), compute risk:

\[
R(t) := \min_{g_t \in G_t} \max_{i \in \text{children}} \frac{c(t)}{F(S) - F(S_{g_t}^i)}
\]

- Choose feature with smallest \( R(t) \)
Main Theorem: 
Trees with Guaranteed Cost

$$Cost_F(S') \leq \log(n) \cdot OPT(S')$$

- max-cost constructed by GreedyTree using $F$ on examples $S$
- number of examples
- minimum max-cost among all trees whose leaves have zero impurity

**Admissible impurity function**

- Nonnegativity: $F(G) \geq 0$ for any set of examples $G$
- Purity: $F(G) = 0$ if $G$ consists of examples of the same class
- Monotonicity: $F(G) \geq F(R), \forall R \subseteq G$
- Supermodular: $F(G \cup j) - F(G) \geq F(R \cup j) - F(R)$ for any $R \subseteq G$ and example $j \notin R$
- Polynomial: $\log(F(S)) = O(\log n)$

Pairs: $F(G) = \sum_{i \neq j} n_i^n j^n$

Gini without normalization

$$R(t) := \min_{g_t \in G_t} \max_{i \in \text{children}} \frac{c(t)}{F(S) - F(S^i_{g_t})}$$
**Proof Sketch**

Let $q$ be such that $Cost_p(S_q^i) = \max_i Cost_p(S^i_q)$.

Claim:

$$\frac{c(\tau)}{F(S) - F(S_{g_{\tau}}^q)} \leq \frac{c_{t_i}}{F(S(v_i)) - F(S(v_{i+1}))}$$

Lower bound $OPT(S)$:

$$\frac{c(\tau)F(S)}{F(S) - F(S_{g_{\tau}}^q)} \leq OPT(S).$$

By induction:

$$\frac{Cost_p(S)}{OPT(S)} \leq \log(F(S)) + 1 = O(\log(n))$$
Experiments

Forest Covertype
[Frank&Asuncion]
- Use less features
- Achieves lower test error

Kusner et al., 2014
Xu et al., 2013
Xu et al., 2012
Yahoo! Learning to Rank [Chapelle et al]

- Less feature extraction cost (CPU time)
- Better ranking accuracy
Conclusion

- Proposed a novel random forest approach for supervised learning under test time budget
- Proposed family of impurity functions such that a simple greedy algorithm produces provably low-cost trees
- Strong empirical performance on real data

Future Work:
- Use out-of-bag samples to improve (prune) trees to save cost
- Guarantee on expected cost of each tree