Recall Tree

Paul Mineiro

December 10, 2016

Motivation

Extreme Classification

- Multiclass with a large number of classes
- Computationally: naive class comparison expensive
- Statistically: many classes have singleton support

One-against-all

- Build a predictor of “this class” vs. “all others”
- OAA is computationally expensive
- Naive OAA generally strong statistically
  - . . . but singleton support is an issue.

Nearest neighbor

- Find closest point and predict that point's class.
  - ...but how to define close?
- Computationally expensive.
  - ...attempt to build an index.
- Competitive when all classes have singleton support.

One-against-some

- Somehow guess which classes are reasonable given the example.
    - To do this, online learn something like an index.
- Run an OAA amongst this small set of candidates.

Recall tree

- Learn a tree structure to route an example to a set of candidate classes.
- Try to increase total probability of top-k classes in each subnode.
- Ex: root node, top-10 classes cover 0.1% of data.
  - left child: top-10 classes cover 0.2% of data.
  - right child; top-10 classes cover 0.15% of data.
  - useful split

Subtleties

- Data decimated as tree depth increases $\Longrightarrow$ overfitting.
  - We have a special regularizer to help mitigate this.
- Path through tree are learned features $\Longrightarrow$ local specialization of class models $\Longrightarrow$ possible to beat OAA.

Demo