Data Mining Lectures - Decision trees

Piotr Wąsiewicz

Institute of Computer Science

pwasiewi@elka.pw.edu.pl

22 maja 2017

Decision trees

- A hierarchical structure representing dataset/domain partitioning nodes: splits based on attribute-value conditions and leaves: class labels or probability distributions.
- Prediction by descending the tree at each node dispatching along a branch at each leaf a class label or probability determined.

Splits for discrete attributes

- Value-based: split outcomes correspond to single attribute values.
- Equality based: split outcomes correspond to binary equality test results.
- Partition-based: split outcomes correspond to attribute value subsets.
- Membership-based: split outcomes correspond to binary membership test results.

Splits for numeric attributes

- Inequality based: split outcomes correspond to binary inequality test results.
- Interval-based: split outcomes correspond to attribute value interval.

Decision tree growing

1. create the root node and mark it as open;
2. assign all training instances from T to the root node;
3. while there are open nodes:
   A. select an open node n;
   B. calculate class distribution $P(d|n)$ based on T[n];
   C. assign class label $argmax[d]P(d|n)$ to n;
   D. if stop criteria are satisfied for n mark n as a closed leaf; else
      1. select a split t for n;
      2. for each outcome r of split t:
      A. create a descendant node n[r] corresponding to r and mark it as open;
      B. assign all instances from T[n,t=r] to n[r];
      C. mark n as a closed node;

Stop criteria

- Uniform class: all training instances in the node are of the same class.
- No instances left: the set of training instances assigned to the node is empty.
- No splits left: there is no split that can be applied to further partition the current subset of training instances.
- Can be relaxed:
  1. most instances of the same class (low class impurity),
  2. less than a specified minimum number of instances,
  3. the best available split is not sufficiently good.

Split selection

- Strict stop criteria guarantee training set error minimization.
- Split selection responsible for overfitting avoidance.
- Ockham's razor: among trees with the same training set error prefer smaller ones and it can be achieved by minimizing class impurity e.g. its entropy.

Pruning and probability classification

- In pruning the complexity parameter (cp) controls the tradeoff between error and size
- Class probability distribution at leaves enables probabilistic prediction
- Can be used to minimize misclassification costs; instead of predicting the most probable class predict the class with the minimum expected cost,
- Can be used to adjust the operating point for binary classification e.g. obtaining the ROC curve