Mining Decision Trees from Data Streams

Tong Suk Man Ivy

CSIS DB Seminar
February 12, 2003

Contents

- Introduction: problems in mining data streams
- Classification of stream data
  - VFDT algorithm
- Window approach
  - CVFDT algorithm
- Experimental results
- Conclusions
- Future work

Data Streams

- Characteristics
  - Large volume of ordered data points, possibly infinite
  - Arrive continuously
  - Fast changing
- Appropriate model for many applications:
  - Phone call records
  - Network and security monitoring
  - Financial applications (stock exchange)
  - Sensor networks

Problems in Mining Data Streams

- Traditional data mining techniques usually require
  - Entire data set to be present
  - Random access (or multiple passes) to the data
  - Much time per data item
- Challenges of stream mining
  - Impractical to store the whole data
  - Random access is expensive
  - Simple calculation per data due to time and space constraints

Classification of Stream Data

- VFDT algorithm
  - "Mining High-Speed Data Streams", KDD 2000.
    Pedro Domingos, Geoff Hulten
- CVFDT algorithm (window approach)
  - "Mining ***Time-Changing*** Data Streams", KDD 2001.
    Geoff Hulten, Laurie Spencer, Pedro Domingos

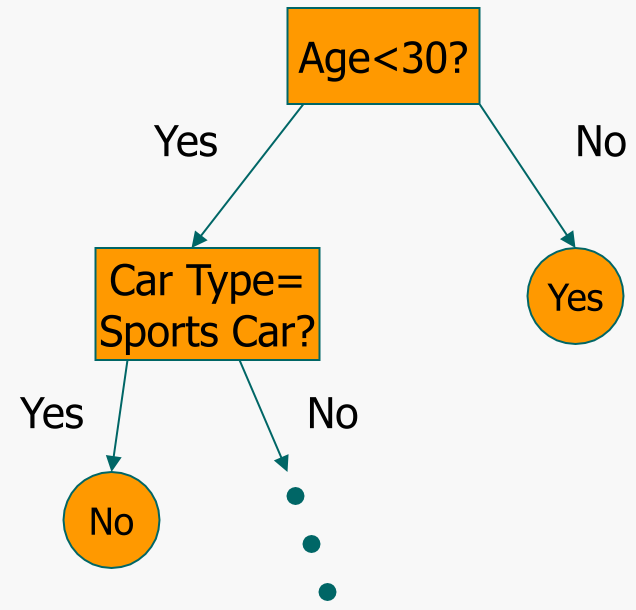
Hoeffding Trees

Definitions

- A classification problem is defined as:
  - N is a set of training examples of the form (x, y)
  - x is a vector of d attributes
  - y is a discrete class label
- Goal: To produce from the examples a model y=f(x) that predict the classes y for future examples x with high accuracy

Decision Tree Learning

- One of the most effective and widely-used classification methods
- Induce models in the form of decision trees
  - Each node contains a test on the attribute
  - Each branch from a node corresponds to a possible outcome of the test
  - Each leaf contains a class prediction
  - A decision tree is learned by recursively replacing leaves by test nodes, starting at the root



Challenges

- Classic decision tree learners assume all training data can be simultaneously stored in main memory
- Disk-based decision tree learners repeatedly read training data from disk sequentially
  - Prohibitively expensive when learning complex trees
- Goal: design decision tree learners that read each example <u>at most once</u>, and use a small constant time to process it

Key Observation

- In order to find the best attribute at a node, it may be sufficient to consider only a small subset of the training examples that pass through that node.
  - Given a stream of examples, use the first ones to choose the root attribute.
  - Once the root attribute is chosen, the successive examples are passed down to the corresponding leaves, and used to choose the attribute there, and so on recursively.
- Use Hoeffding bound to decide how many examples are enough at each node

Hoeffding Bound

- Consider a random variable a whose range is R
- Suppose we have n observations of a
- Mean: $\bar{a}$
- Hoeffding bound states:

With probability 1- δ, the true mean of a is at least $\bar{a} - \varepsilon$, where $$\varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$$

How many examples are enough?

- Let G(X_i) be the heuristic measure used to choose test attributes (e.g. Information Gain, Gini Index)
- X_a : the attribute with the highest attribute evaluation value after seeing n examples.
- X_b : the attribute with the second highest split evaluation function value after seeing n examples.
- Given a desired δ, if $\Delta\overline{G} = \overline{G}(X_a) - \overline{G}(X_b) > \varepsilon$ after seeing n examples at a node,
  - Hoeffding bound guarantees the true $\Delta G >= \Delta\overline{G} - \varepsilon > 0$, with probability $1-\delta$.
  - This node can be split using X_a, the succeeding examples will be passed to the new leaves.

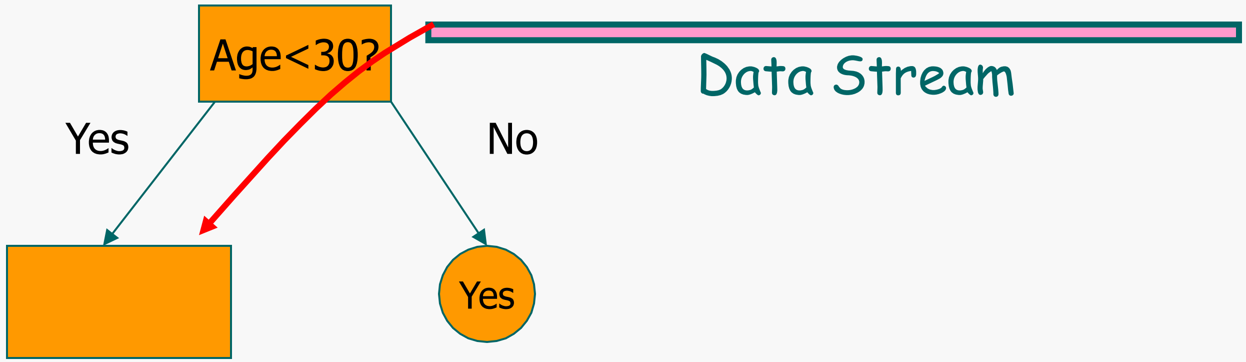
$$\varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$$

Algorithm

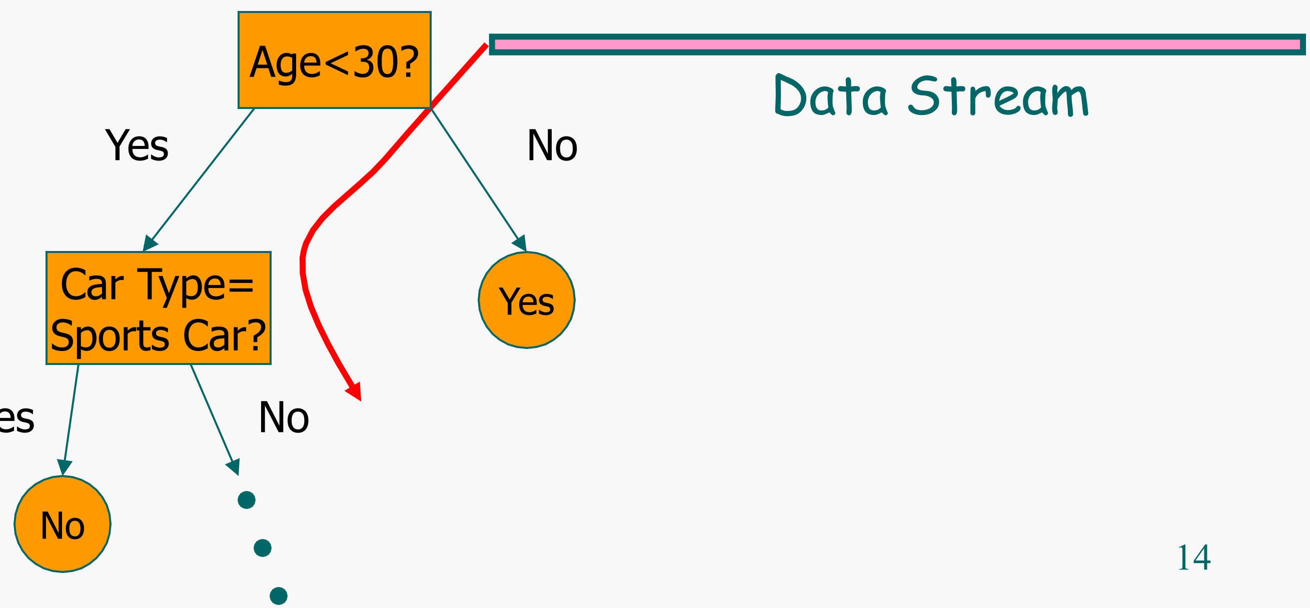
- Calculate the information gain for the attributes and determines the best two attributes
  - Pre-pruning: consider a “null” attribute that consists of not splitting the node
- At each node, check for the condition

$$\Delta\overline{G} = \overline{G}(X_a) - \overline{G}(X_b) > \varepsilon$$

- If condition satisfied, create child nodes based on the test at the node
- If not, stream in more examples and perform calculations till condition satisfied



$\bar{G}(\text{Car Type}) - \bar{G}(\text{Gender}) > \varepsilon$



Performance Analysis

- p: probability that an example passed through DT to level i will fall into a leaf at that point
- The expected disagreement between the tree produced by Hoeffding tree algorithm and that produced using infinite examples at each node is no greater than ♎/p.
- Required memory: O(leaves * attributes * values * classes)

VFDT

VFDT (Very Fast Decision Tree)

- A decision-tree learning system based on the Hoeffding tree algorithm
- Split on the current best attribute, if the difference is less than a user-specified threshold
  - Wasteful to decide between identical attributes
- Compute G and check for split periodically
- Memory management
  - Memory dominated by sufficient statistics
  - Deactivate or drop less promising leaves when needed
- Bootstrap with traditional learner
- Rescan old data when time available

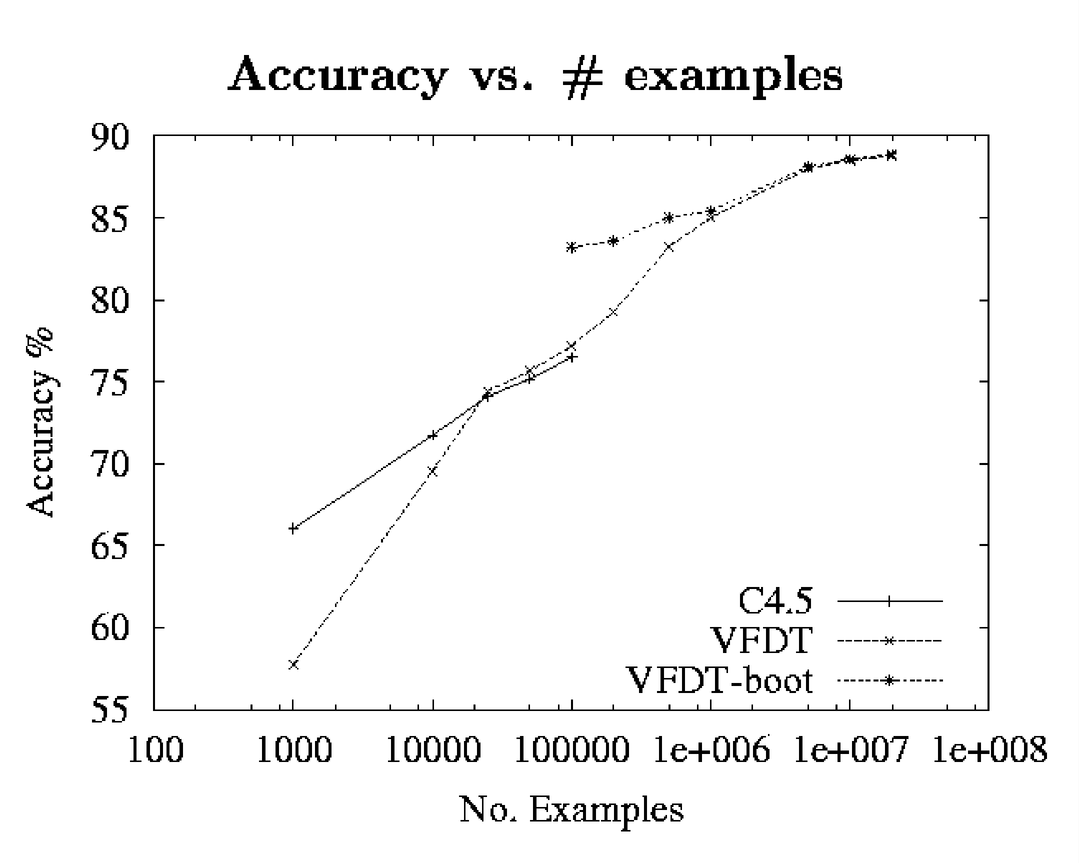
VFDT(2)

- Scales better than pure memory-based or pure disk-based learners
  - Access data sequentially
  - Use subsampling to potentially require much less than one scan
- VFDT is incremental and anytime
  - New examples can be quickly incorporated as they arrive
  - A usable model is available after the first few examples and then progressively defined

Experiment Results (VFDT vs. C4.5)

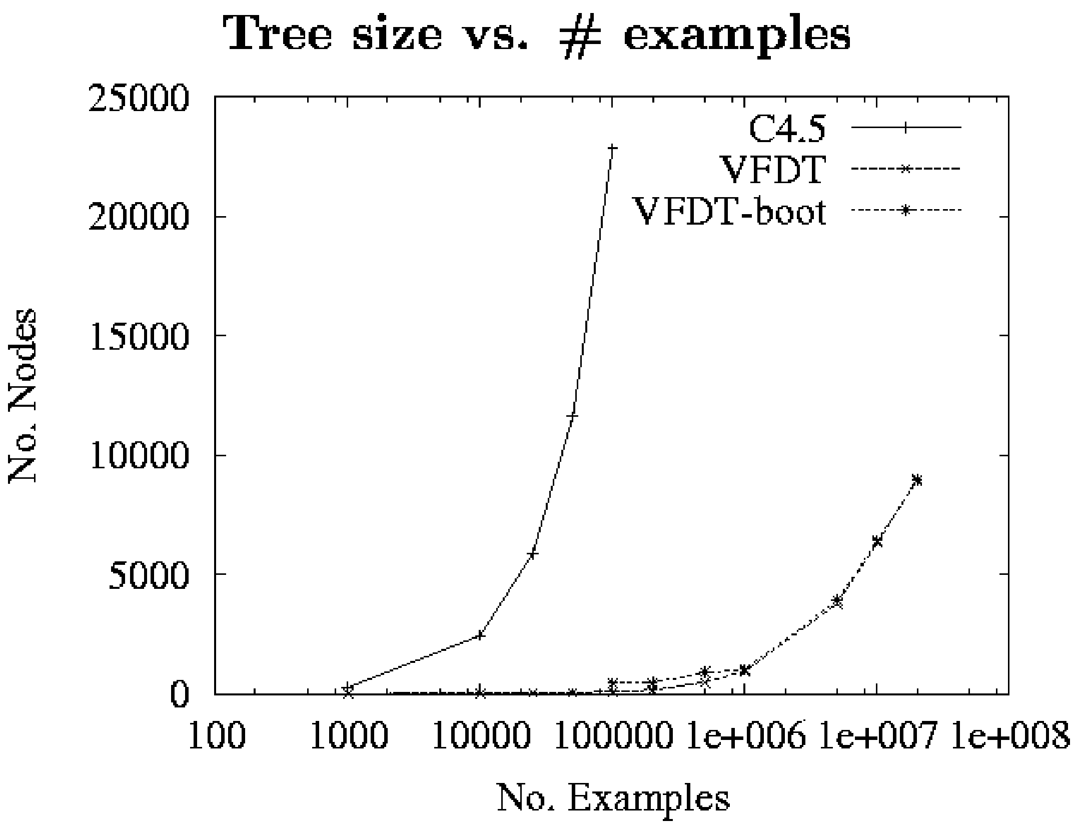
- Compared VFDT and C4.5 (Quinlan, 1993)
- Same memory limit for both (40 MB)
  - 100k examples for C4.5
- VFDT settings: $\delta = 10^{-7}$, $\tau = 5\%$, $n_{min}=200$
- Domains: 2 classes, 100 binary attributes
- Fifteen synthetic trees 2.2k – 500k leaves
- Noise from 0% to 30%

Experiment Results



Accuracy as a function of the number of training examples

Experiment Results



Tree size as a function of number of training examples

Mining Time-Changing Data Stream

- Most KDD systems, include VFDT, assume training data is a sample drawn from stationary distribution
- Most large databases or data streams violate this assumption
  - Concept Drift: data is generated by a time-changing concept function, e.g.
    - Seasonal effects
    - Economic cycles
- Goal:
  - Mining continuously changing data streams
  - Scale well

Window Approach

- Common Approach: when a new example arrives, reapply a traditional learner to a sliding window of w most recent examples
  - Sensitive to window size
    - If w is small relative to the concept shift rate, assure the availability of a model reflecting the current concept
    - Too small w may lead to insufficient examples to learn the concept
  - If examples arrive at a rapid rate or the concept changes quickly, the computational cost of reapplying a learner may be prohibitively high.

CVFDT

CVFDT

- CVFDT (Concept-adapting Very Fast Decision Tree learner)
  - Extend VFDT
  - Maintain VFDT's speed and accuracy
  - Detect and respond to changes in the example-generating process