

Data Mining on Streams using Decision Trees

CS 536: Machine Learning

Instructor: Michael Littman

TA: Yihua Wu

1

Outline

- Introduction to Data Streams
 - Motivation
 - Models
- Summarizing data streams
 - Sampling-based
 - Sketch-based
- Overview of traditional DT learning Algs
- DT learning Algs on streams
 - One concept
 - Multiple concepts (concept drifts)

2

Puzzle: Find missing numbers

- **Paul** permutes numbers $1..n$, and shows all but one to Carol, in the permuted order, one after the other.
- **Carol** must find the missing number.

Carol can not remember all the numbers she has been shown.

Carol finds the missing number...

- Carol **cumulates** the sum of all the numbers she is being shown. At the end, she can subtract this sum from the total sum of the numbers $1..n$.
 - Uses $O(\log n)$ bits to **store** the partial sum.
 - Performs one + **each time** Paul shows a number. Takes $O(\log n)$ time per number.
 - At the end, computes the missing number with one subtraction. Takes $O(\log n)$ time for **final computation**.

Finding two missing numbers...

- What if Paul shows all but **two** numbers?
- Carol keeps the **sum AND product** of the numbers Paul shows her.
 - $O(\log n!) = O(n \log n)$ bits and time.
- Alternatively, Carol keeps the **sum AND sum of squares** of the numbers Paul shows her.
 - As before: $O(\log n)$ storage,
 $O(\log n)$ process time and
 $O(\log n)$ compute time.

5

Desiderata

- $Polylog(n)$ per item processing time
- $Polylog(n)$ space stored
- $Polylog(n)$ for computing functions on s
- Can only scan stream **once!**

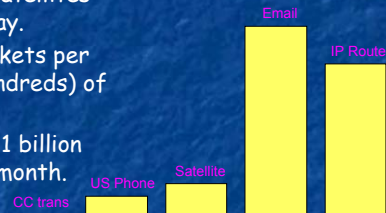
	Traditional	Stream
num of passes	multiple	single
time	unlimited	polylog
memory	unlimited	polylog
result	accurate	approximate
num of concepts	one	multiple

6

The Data Stream Phenomenon

- Highly detailed, automatic, rapid data feeds.

- 3 Billion Telephone Calls in US each day
- 30 Billion emails daily, 1 Billion SMS
- Scientific data:** NASA's observation satellites generate billions of readings each per day.
- IP Network Traffic:** up to 1 Billion packets per hour per router. Each ISP has many (hundreds) of routers!
- Compare to "human scale" data: "only" 1 billion worldwide credit card transactions per month.



- Need for near-real time analysis of data feeds. (classification; extreme events—heavy hitters, deltoids; etc.)

Models of Data Streams

- Signal $s[1..n]$. n is universe size.
- Three models:
 - Time-series model:** $s(1), s(2), \dots, s(t), \dots$
 - Cash Register model:** $s_t(j) = s_{t-1}(j) + a_j(t)$. $a_j(t) > 0$. (insert only)
 - Turnstile model:** $s_t(j) = s_{t-1}(j) + u_j(t)$. (both insertion and deletion)

Summarizing Data Streams

Synopsis

- A small space representation of the data stream
- Can be rapidly updated on the fly
- Approx results: within error bounds with high probability

Two approaches

- **Samples**: use a subset to present all
- **Sketches**: all records observed, but store using much less space (e.g. sum)

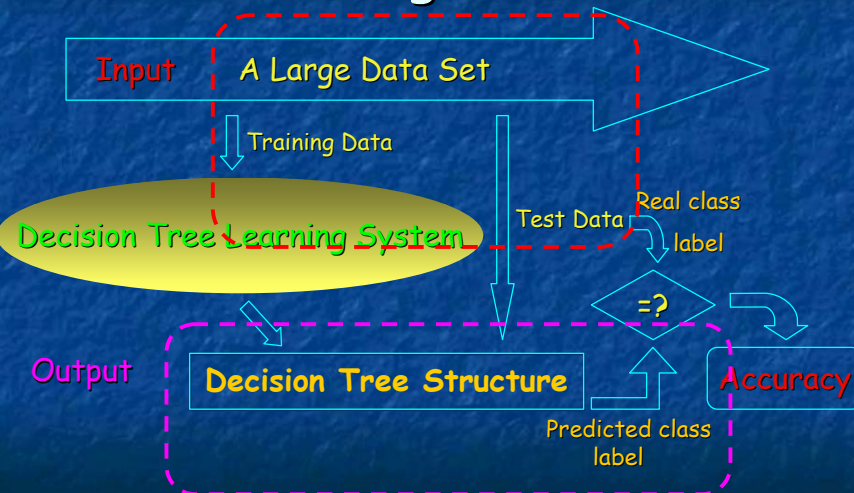
9

What is a Decision Tree?

- **Decision Tree** is a classification model for finding patterns in data using tree structures. Can be used to explain data and make predictions from it.
 - **Internal Nodes**: tests on examples' attribute values.
 - **Leaf Nodes**: class labels.
- Applies to **categorical** outputs.
- **ID3**: a DTLA, only **categorical** attributes.

10

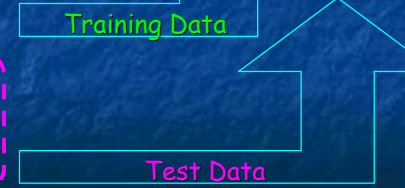
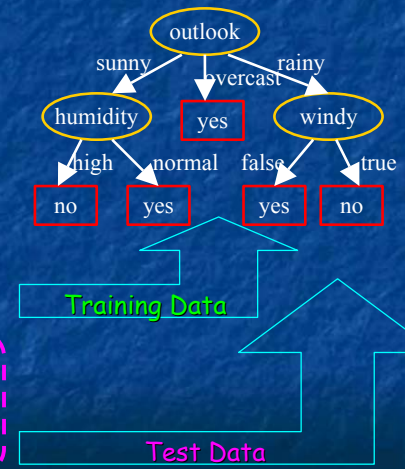
Learning Process



Here is an Example...

outlook temp humidity windy play

sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no



Building Decision Trees

Key Idea

- Evaluate splits for each attribute;
- Pick the best attribute to test at root;
- Divide the training data into subsets D_i for each value the attribute can take on;
- Recurse the tree construction for each D_i .

Attribute Selection

- Find an attribute that divides data into as **pure** subsets as possible.

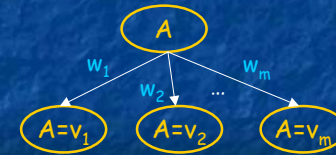
13

Information Gain

- C has r possible values, and A has m .
- For any dataset D ,

$$\text{Split-Info}(A, D) = \sum_{i=1}^m w_i * H\left(\frac{|D_{A=v_i, C=c_1}|}{|D|}, \frac{|D_{A=v_i, C=c_2}|}{|D|}, \dots, \frac{|D_{A=v_i, C=c_r}|}{|D|}\right)$$

$$IG(A, D) = H\left(\frac{n_1}{n}, \frac{n_2}{n}, \dots, \frac{n_m}{n}\right) - \text{Split-Info}(A, D), \text{ where } w_i = \frac{|D_{A=v_i}|}{|D|}$$

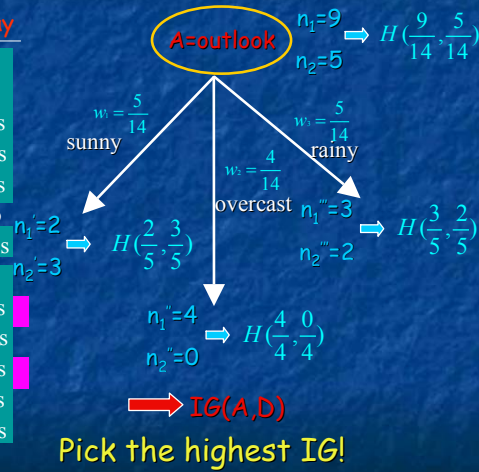


14

Information Gain, cont'd

outlook temp humidity windy play

sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no



Sufficient Statistics

- Recall that ...

$$\text{Split-Info}(A, D) = \sum_{i=1}^m w_i * H\left(\frac{|D_{A=v_i, C=c_1}|}{|D|}, \frac{|D_{A=v_i, C=c_2}|}{|D|}, \dots, \frac{|D_{A=v_i, C=c_r}|}{|D|}\right)$$

$$= \sum_{i=1}^m \frac{|n_i|}{|n|} \sum_{k=1}^r \left(-\frac{n_{ik}}{n_{i\cdot}} \log_2 \frac{n_{ik}}{n_{i\cdot}}\right)$$

- Sufficient Statistic:** n_{ijk} is the number of examples whose i^{th} attribute takes the j^{th} value, and are classified to the k^{th} class.
 - Total # of n_{ijk} 's = $m * l * R$.

Drawbacks

- One pass of data for each layer, multiple passes in total. (**stream: only one pass**)
- Once make a decision on a splitting attribute, never reconsider. (**stream: concept drifts**)

17

Hoeffding Bound

- Consider a real-valued random variable r whose range is R . Suppose we have n independent observations of this variable, and compute their mean $\text{mean}(r)$. The hoeffding bound states that, with probability $1-\delta$, the true mean of the variable is at least $\text{mean}(r) - \epsilon$, where

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

18

Properties

- The hoeffding bound is independent of the probability distribution generating the observations.
- With high probability, the attribute chosen using n examples is the same that would be chosen using infinite examples.

Hoeffding Tree Algorithm

Mining High-Speed Data Streams, Pedro Domingos and Geoff Hulten, KDD 2000

- **Inputs:** S is a sequence of examples,
 X is a set of categorical attributes,
 G is a split evaluation function,
 δ is one minus the desired probability of choosing the correct attribute at any given node.
- **Output:** HT is a decision tree.

Hoeffding Tree Algorithm, cont'd

Procedure HoeffdingTree($S, X, \mathcal{G}, \delta$)

Let HT be a tree with a single leaf l_1 (the root).

For each class y_k
For each value x_{ij} of each attribute X_i in X
Let $n_{ijk}(l_1)=0$. } **init SS's**

For each example (x, y_k) in S
Sort (x, y) into a leaf l using HT.
For each x_{ij} in x such that X_i in X_l
Increment $n_{ijk}(l)$. } **update SS's**

If the examples seen so far at l are not all of the same class, then

Compute $G_i(X_i)$ for each attribute X_i in X_l using $n_{ijk}(l)$.

Let X_a be the attribute with highest G_i .

Let X_b be the attribute with second-highest G_i .

Compute ϵ using hoeffding bound.

If $G_i(X_a) - G_i(X_b) > \epsilon$, then

Replace l by an internal node that splits on X_a .

For each branch of the split

Add a new leaf l_m , and let $X_m = X - \{X_a\}$.

For each class y_k and each value x_{ij} of each attribute X_i in X_m

Let $n_{ijk}(l_m)=0$. } **evaluate functions**

Return HT.

21

Problems in Practice

- More than one attribute very close to the current best.
- How much time spent on a single example?
- Memory needed with the tree expansion?
- Number of candidate attributes at each node?

22

VFDT System

- It's a **Very Fast Decision Tree** learner, based on the hoeffding tree algorithm.
- **Refinements:**
 - **Ties.** If $\Delta G < \epsilon < \tau$, where τ is a user-specified threshold, split on the current best attribute.
 - **G computation.** Specify an n_{min} that must be accumulated at a leaf before G is recomputed.
 - **Memory.** If the max available memory is reached, VFDT deactivates the least promising leaves (w/ the lowest p_{e_i}) to make room for new ones. Can be reactivated if more promising later.
 - **Poor attributes.** Memory is minimized by dropping early on attributes whose difference from the best attribute's G becomes greater than ϵ .

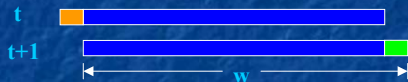
VFDT Analysis

- **Memory:** $O(ldvc)$
 - l : the number of leaves in the tree
 - d : the number of attributes
 - v : the max number of values per attribute
 - c : the number of classes

It's independent of the number of examples seen.
- **Drawback:** doesn't take care of the time-changing data streams, because we never update the tree structure ever since we finish building the tree.

Brute Force Algorithm

- A sliding window + the VFDT
 - Reapply VFDT to a moving window of examples every time a new example arrives.



- From $t \rightarrow t+1$, only $O(1)$ item in the sliding window changes, but we have to rescan $O(w)$ items if reapply VFDT.

25

CVFDT

Mining Time-Changing Data Streams, Geoff Hulten, Laurie Spencer, and Pedro Domingos, KDD 2001

■ Concept-adapting Very Fast Decision Tree

■ **Basic Ideas:**

- An extension to VFDT.
- Maintains VFDT's speed and accuracy.
- Detects and responds to concept changes in $O(1)$ per example.
- Stays current while making the most of old data by growing an alternative subtree whenever an old one becomes questionable.
- And replace the old with the new when the new becomes more accurate.

26

CVFDT Algorithm

- **Tree node** (internal node & leaf node of HT and all alternate trees)
 - maintain sufficient statistics n_{ijk}
 - assigned a unique, monotonically increasing ID when created.
- **Sliding window**
 - the max ID of the leaves an example reaches is attached with the example in W .

CVFDT Algorithm, cont'd

- **observe a new example**
 - increase the sufficient statistics n_{ijk} along the way from the root to leaves.
 - record the max ID of the leaves it reaches in HT and all alternate trees.
- **forget the old**
 - decrease the sufficient statistics n_{ijk} of every node the example reaches whose ID \leq the stored ID.

CVFDT Algorithm, cont'd

- **Growth of alternate subtrees**
 - If $G(X_a) - G(X_b) \leq \epsilon$ and $\epsilon > \tau$, grow a subtree.
 - Check periodically, say every f examples.
- **Replacement with alternate subtrees**
 - The next coming m examples are used to compare the accuracy of the current subtree in HT with the accuracies of all of its alternate subtrees.
 - Replace if the most accurate alternate is more accurate than the current.
 - Prune alternate subtrees that are not making progress.
 - Check periodically.

29

CVFDT vs. VFDT

	VFDT	CVFDT
Memory	$O(ldvc)$	$O(ndvc)$
Time	$O(l_v dvc)$	$O(l_c dvc)$

- l : the number of leaves in HT
- n : the number of nodes in the main tree and all alternate subtrees
- d : the number of attributes
- v : the max number of values per attribute
- c : the number of classes
- l_v : the height of HT
- l_c : the length of the longest path through HT times the number of alternate trees

30

What if ...

- The number of attribute values is huge. e.g. # of IP addresses $n = 2^{32}$.
- The flavor of sketch-based solution...



- Hash functions and statistical techniques needed to guarantee accuracy and efficiency.
- Why not approx n_{ijk} 's? **more time efficient!**

References

- Mining High-Speed Data Streams, Pedro Domingos and Geoff Hulten, KDD 2000
- Mining Time-Changing Data Streams, Geoff Hulten, Laurie Spencer, and Pedro Domingos, KDD 2001
- A survey on Data Stream Algorithms. S. Muthukrishnan

Thanks!



Q & A ???