Mining from Data Streams: Decision Trees

João Gama
LIAAD-INESC Porto,
University of Porto, Portugal
jgama@fep.up.pt

Outline

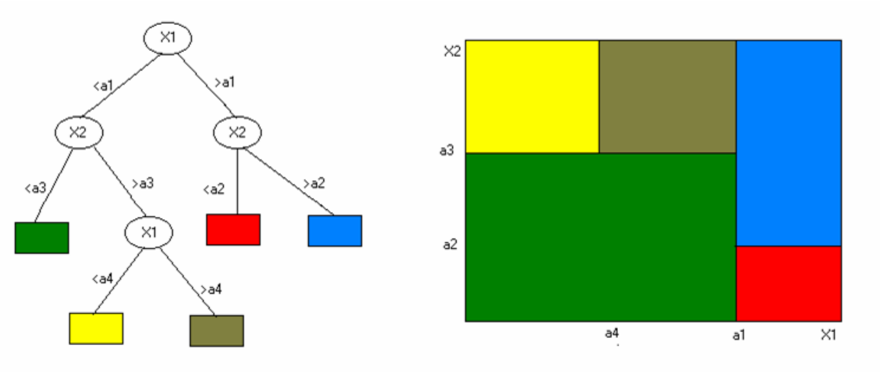
- A decision tree uses a divide-and-conquer strategy:
    - A complex problem is decomposed into simpler sub problems.
    - Recursively the same strategy is applied to the sub problems.
- The discriminant capacity of a decision tree is due to:
    - Its capacity to split the instance space into sub spaces.
    - Each sub space is fitted with a different function.
- There is increasing interest:
    - CART (Breiman, Friedman, et.al.)
    - C4.5 (Quinlan)
    - Splus, Statistica, SPSS, R, ...
    - IBM IntelligentMiner, Microsoft SQL Server, ...

Decision Trees

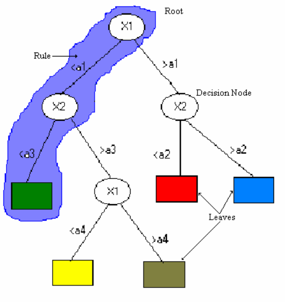
Decision trees are one of the most commonly used algorithms, on both in real world applications and in academic research.

- *Flexibility:* Non-parametric method.
- *Robustness:* Invariant under all (strictly) monotone transformations of the individual input variables.
- *Feature Selection:* Robust against the addition of irrelevant input variables.
- *Interpretability:* Global and complex decisions can be approximated by a series of simpler and local decisions.
- *Speed:* Greedy algorithms that grows top-down using a divide-and-conquer strategy without backtracking.

Partition of the Instance Space

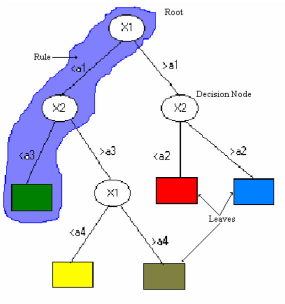


Representation of a Decision Tree



- Representation using decision trees:
  - Each decision node contains a test in one attribute
  - Each descendant branch correspond to a possible attribute-value.
  - Each terminal node (leaf) predicts a class label.
  - Each path from the root to the leaf corresponds to a classification rule.

Decision Tree Representation



- In the attribute space:
  - Each leaf corresponds to a decision region (Hyper-rectangle)
  - The intersection of the hyper-rectangles is Null
  - The union of the hyper-rectangles is the universe.

Decision Tree Representation

A Decision Tree represents a disjunction of conjunctions of restrictions in the attribute values.

- Each branch in a tree corresponds to a conjunction of conditions.
- The set of branches are disjunct.
- DNF (disjunctive normal form)

Learning from Data Streams: Desirable Properties

- Processing each example:
  - Small constant time
  - Fixed amount of main memory
  - Single scan of the data
    without (or reduced) revisit old records.
  - Processing examples at the speed they arrive
- Ability to detect and react to concept drift
- Decision Models at anytime
- Ideally, produce a model equivalent to the one that would be obtained by a batch data-mining algorithm

Incremental Decision Trees I

Algorithms using tree re-structuring operators.
When new information is available splitting-tests are re-evaluated

- Incremental Induction of Topologically Minimal Trees
  Walter Van de Velde, 1990
- Sequential Inductive Learning
  J.Gratch, 1996
- Incremental Tree Induction
  P.Utgoff, 1997
- Efficient Incremental Induction of Decision Trees
  D.Kalles, 1995

Incremental Decision Trees II

Algorithms that do not re-consider splitting-test changes.
Install a splitting test only when there is evidence enough in favor to that test.

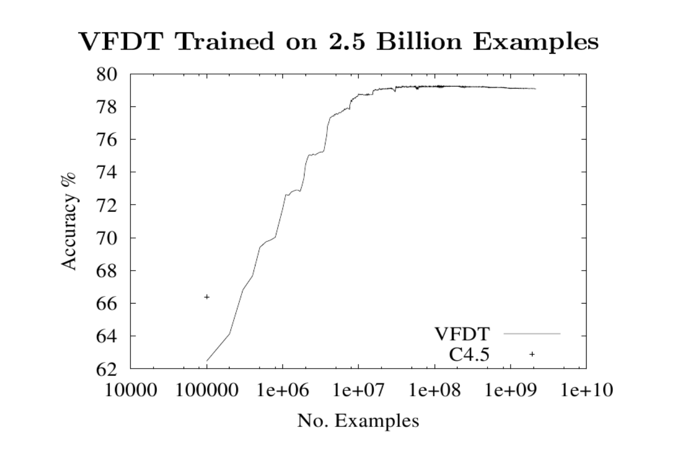
- Very Fast Decision Tree (VFDT)
  P.Domingos, KDD, 2000
- Very Fast Decision Tree for Continuous Attributes(VFDTc)
  J. Gama, KDD, 2003
- Ultra-Fast Decision Trees (UFFT)
  J. Gama, Sac04

Outline

Do you need so many examples ?

Domingos, Hulten: *Mining High Speed Data Streams*, KDD00

VFDT: Illustrative Evaluation – Accuracy



Very Fast Decision Trees

The base Idea

A small sample can often be enough to choose the optimal splitting attribute

- Collect sufficient statistics from a small set of examples
- Estimate the merit of each attribute

How large should be the sample?

- **The wrong idea:** Fixed sized, defined *apriori* without looking for the data;
- **The right idea:** Choose the sample size that allow to differentiate between the alternatives.

Very Fast Decision Trees

Mining High-Speed Data Streams, P. Domingos, G. Hulten; KDD00

The base Idea

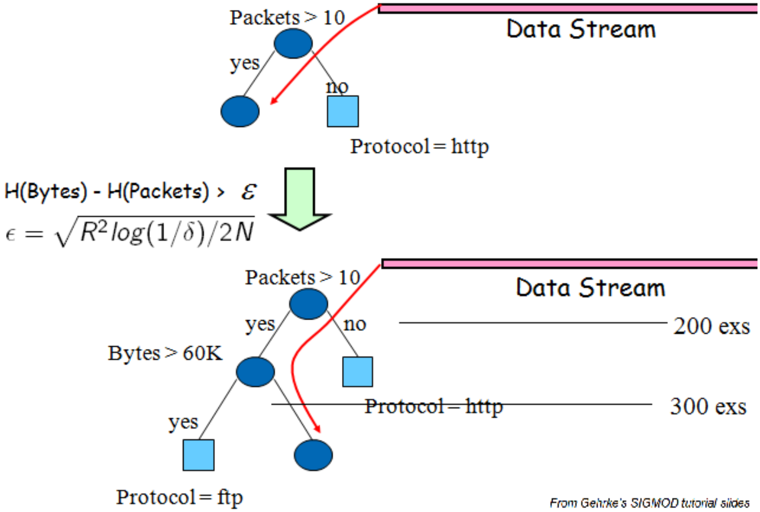
A small sample can often be enough to choose the optimal splitting attribute

- Collect sufficient statistics from a small set of examples
- Estimate the merit of each attribute
- Use Hoeffding bound to guarantee that the best attribute is really the *best*.
  - Statistical evidence that it is better than the second best

Very Fast Decision Trees: Main Algorithm

- **Input:** δ desired probability level.
- **Output:** $\mathcal{T}$ A decision Tree
- **Init:** $\mathcal{T} \leftarrow$ Empty Leaf (Root)
- While (TRUE)
  - Read next example
  - Propagate example through the tree from the root till a leaf
  - Update sufficient statistics at leaf
  - If $leaf(\#examples) > N_{min}$
    - Evaluate the merit of each attribute
    - Let A_1 the best attribute and A_2 the second best
    - Let $\epsilon = \sqrt{R^2 ln(1/\delta)/(2n)}$
    - If $G(A_1) - G(A_2) > \epsilon$
    - Install a splitting test based on A_1
    - Expand the tree with two descendant leaves

VFDT


Packets > 10
yes
no
Data Stream
Protocol = http
H(Bytes) - H(Packets) > ε
$\epsilon = \sqrt{R^2 log(1/\delta)/2N}$
Packets > 10
yes
no
Data Stream
200 exs
Bytes > 60K
Protocol = http
300 exs
yes
Protocol = ftp
From Gehrke's SIGMOD tutorial slides

Evaluating the merit of an Attribute

How to choose an attribute?

How to measure the ability of an attribute to discriminate between classes?

Many measures

There are many measures. All agree in two points:

- A split that maintains the class proportions in all partitions is useless.
- A split where in each partition all examples are from the same class has maximum utility.

Entropy

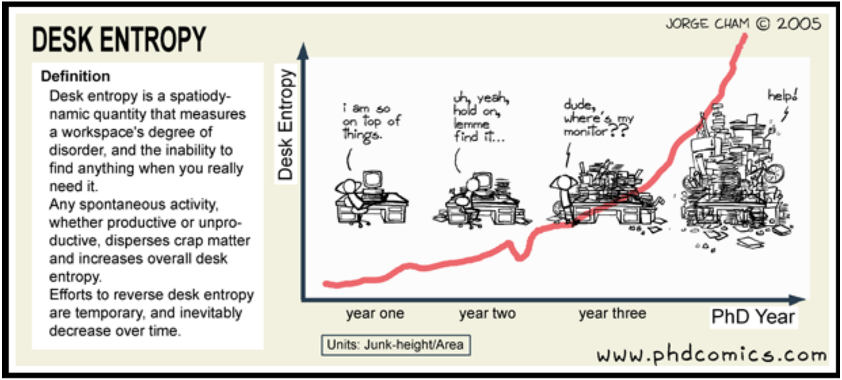
Entropy measures the degree of randomness of a random variable.
The entropy of a discrete random variable which domain is $\{V_1, ...V_i\}$:

$$H(X) = -\sum_{j=1}^{i} p_j log_2(p_j)$$

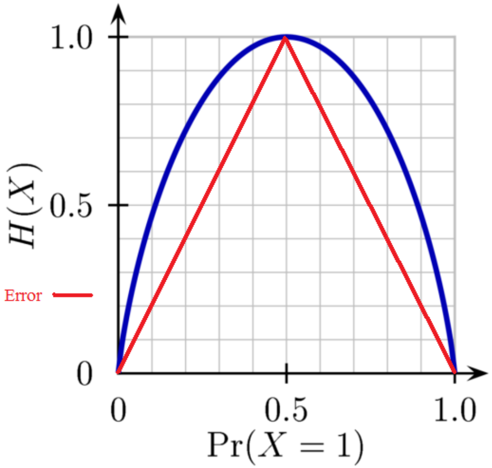
where p_j is the probability of observing value V_j.
Properties:

- $H(X) \geq 0$
- Maximum: $max(H(X)) = log_2 i$ iff $p_i = p_j$ for each $i, j, i \neq j$.
- Minimum: $H(X) = 0$ if there is i such that $p_i = 1$ assuming $0 * log_2 0 = 0$.

Entropy


DESK ENTROPY
Definition
Desk entropy is a spatiodynamic quantity that measures a workspace's degree of disorder, and the inability to find anything when you really need it.
Any spontaneous activity, whether productive or unproductive, disperses crap matter and increases overall desk entropy.
Efforts to reverse desk entropy are temporary, and inevitably decrease over time.
JORGE CHAM © 2005
Desk Entropy
i am so on top of things.
uh, yeah, hold on, lemme find it...
dude, where's my monitor??
help!
year one
year two
year three
PhD Year
Units: Junk-height/Area
www.phdcomics.com

Entropy



Let p_i be the probability that an arbitrary example in D belongs to class C_i, estimated by $|C_i, D|/|D|$

Expected information (entropy) needed to classify an example in D: $H(D) = -\sum p_i \times log_2(p_i)$

Information needed (after using A to split D into v partitions) to classify D: $H_A(D) = \sum_1^v \frac{|D_j|}{|D|} \times H(D_j)$

Information gained by branching on attribute A: $Gain_A = H(D) - H_A(D)$.

Decision Trees and Entropy

Entropy is used to estimate the randomness or difficulty to predict, of the target attribute.

Splitting Criteria

How many examples we need to expand a leaf?
After processing a small sample, Let

- $G(A_1)$ be the merit of the best attribute
- $G(A_2)$ the second best attribute

Question:

Is A_1 a stable option?
what if we observe more examples?

Hoeffding bound

- Suppose we have made n independent observations of a random variable r whose range is R.
  Let $\overline{r}$ be the mean computed in the sample.
- The Hoeffding bound states that:
  - With probability $1 - \delta$
  - The true mean of r is in the range $\overline{r} \pm \epsilon$ where $\epsilon = \sqrt{\frac{R^2 ln(1/\delta)}{2n}}$
- Independent of the probability distribution generating the examples.

Hoeffding bound

- The heuristic used to choose test attributes is the information gain $G(.)$
- Select the attribute that maximizes the information gain.
- The range of information gain is $log(\#classes)$
- Suppose that after seeing n examples, $G(X_a) > G(X_b) > ... > G(X_k)$
- Given a desired ϵ, the Hoeffding bound ensures that X_a is the correct choice, with probability $1 - \delta$, if $G(X_a) - G(X_b) > \epsilon$.

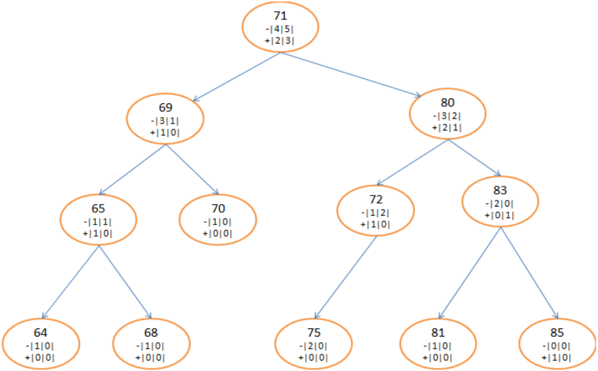
VFDT: Sufficient Statistics

Each leaf stores sufficient statistics to evaluate the splitting criterion

What are the sufficient Statistics stored in a Leaf?

- For each attribute
  - If Nominal
    - Counter for each observed value per class
  - If Continuous
    - Binary tree with counters of observed values
    - Discretization: e.g. 10 bins over the range of the variable
    - Univariate Quadratic Discriminant (UFFT)

Growing a Btree



Computing the Gain for Continuous Attributes

- Each leaf contains a Btree for each continuous attribute
- Traversing the Btree once, it is possible to estimate the gain of all possible cut-points of the attribute
- A cut-point is each observed value in the examples at that leaf

Cut-point	71		69		65		64		68		70		80		72		75		83		81		85	
Classes	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
0	4	5	3	6	1	8	1	8	2	7	4	5	7	2	5	4	7	2	9	0	8	1	9	0
1	2	3	1	4	1	4	0	5	1	4	1	4	4	1	3	2	3	2	4	1	4	1	5	0
total	6	8	4	10	2	12	1	13	3	11	5	9	11	3	8	6	10	4	13	1	12	2	14	0

Computing Information gain for cut-point=81 :

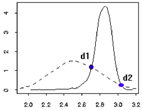
$$\inf o(T_0) = -\frac{8}{12} \times \log_2\left(\frac{8}{12}\right) - \frac{4}{12} \times \log_2\left(\frac{4}{12}\right) = 0.92 \ \ bits.$$

$$\inf o(T_1) = -\frac{1}{2} \times \log_2\left(\frac{1}{2}\right) - \frac{1}{2} \times \log_2\left(\frac{1}{2}\right) = 1 \ \ bit.$$

$$\inf o_{z\text{-}Temperatura}(T) = \frac{12}{14} \times 0.92 + \frac{2}{14} \times 1 = 0.93 \ \ bits.$$

UFFT: Univariate Discriminant Analysis.

- All candidate splits will have the form of $Attribute_i \leq value_j$
- For each attribute, quadratic discriminant analysis defines the cut-point.
- Assume that for each class the attribute-values follows a univariate normal distribution $N(\bar{x}_i, \sigma_i)$.
- The best cut-point is the solution of: $P(+)N(\bar{x}_+, \sigma_+) = P(-)N(\bar{x}_-, \sigma_-)$
- A quadratic equation with at most two solutions: d_1, d_2
- The solutions of the equation split the X-axis into three intervals: $]-\infty, d_1], [d_1, d_2], [d_2, +\infty[$
- We choose between d_1 or d_2, the one that is closer to the sample means.



VFDTc - Missing Values

- Learning Phase:
  - The sufficient statistics of an attribute are not updated whenever a missing value is observed.
- Whenever an example traverse the tree
  - If the splitting attribute is missing in the example, it is locally replaced with:
    - Nominal: the mode of observed values.
    - Continuous: the mean of observed values.
  - These statistics are computed and stored when a leaf is expanded.

Outline

Classification Strategies

- To classify an unlabeled example:
  - The example traverses the tree from the root to a leaf
  - It is classified using the information stored in that leaf

Vfdt like algorithms store in leaves much more information:

- The distribution of attribute values per class.
- Required by the splitting criteria
- Information collected from hundred's (or thousand's) of examples!

How can we use this information?

Functional Leaves

- CART book (Breiman, Freadman, et al)
  grow a small tree using only the most significant splits. Then do multiple regression in each of the terminal nodes.
- Perceptron trees
  P. Utgoff, 1988
- NBTree
  R. Kohavi, 1996
- Hybrid decision tree learners
  A. Seewald, 2001
- Functional Trees, Machine Learning, 2004
  J. Gama
- ...

Classification Strategies

Accurate Decision Trees for mining high-speed Data Streams, J.Gama, R. Rocha; KDD03

Two classification strategies:

- The standard strategy use ONLY information about the class distribution: $P(Class_i)$
- A more informed strategy, use the sufficient statistics $P(x_j|Class_i)$
  - Classify the example in the class that maximizes $P(C_k|\overrightarrow{x})$
  - Naive Bayes Classifier: $P(C_k|\overrightarrow{x}) \propto P(C_k)\prod P(x_j|C_k)$.
    - VFDT stores sufficient statistics of hundred of examples in leaves.

Functional Leaves in VFDTc

VFDTc classifies test examples using a naive Bayes algorithm

Why Naive Bayes?

- NB can use all the information available at leaves
- Is Incremental by nature.
- Process heterogeneous data, missing values, etc.
- Can use the splitting criteria sufficient statistics
- NB is very competitive for small data sets.

VFDTc: Classifying a Test Example

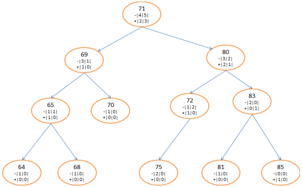
Suppose a test example: $\vec{x} = \{a_1, \ldots, a_n\}$
Naive Bayes formula: $P(C_k|\overrightarrow{x}) \propto P(C_k) \prod P(x_j|C_k)$.
We need to estimate

- The prior probability for each class: $P(C_k)$;
- The conditional probabilities of each attribute-value given the class $P(a_j = i|C_k)$

VFDTc: Classifying a Test Example

- Nominal Attributes:
  - Conditional probabilities: $P(a_j = j|k) = n_{ijk}/n_k$
  - Already stored in leaves
- Continuous Attributes:
  - Supervised discretization:
    Number of bins: min(10, nr. Of distinct observed values).
  - Equal-width bins
  - Defining the breaks is trivial given:
    - The range of the attribute and
    - the number of bins
  - How to fill in bins?
    Traversing the Btree once!

VFDTc: Classifying a Test Example

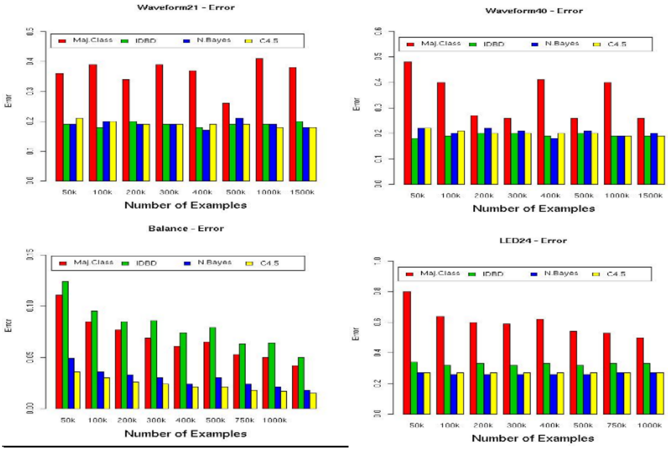


Traversing the Btree once:

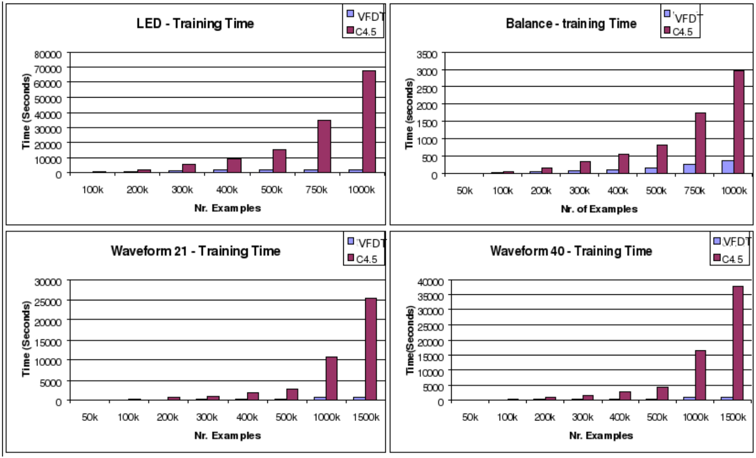
- The range of the variable at that node;
- The Contingency Table

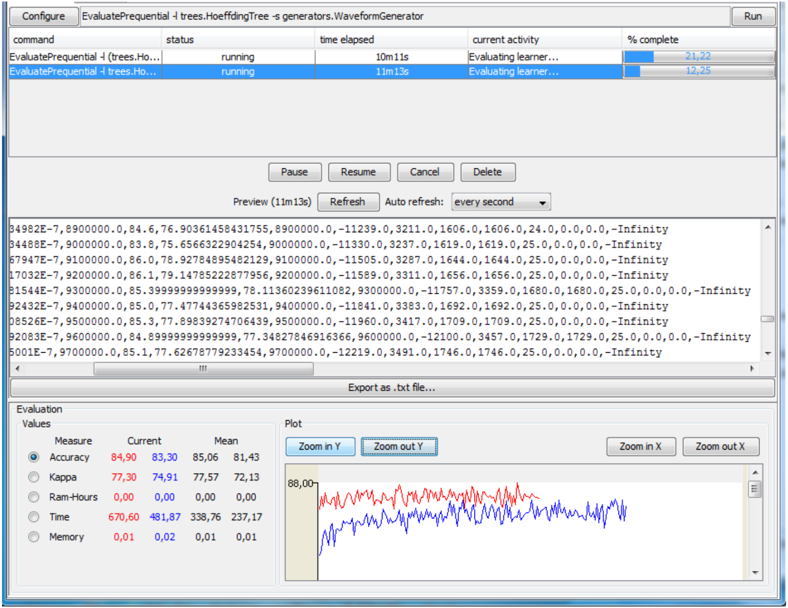
Interval Classes	]-,66.1]	]66.1,68.2]	]68.2,70.3]	]70.3,2.4]	]72.4,74.5]	]74.5,76.6	]76.6,78.7]	]78.7,80.8]	]80.8,82.9]	]82.9,+[
0	1	1	2	1	0	2	0	0	1	1
1	1	0	0	2	0	0	0	1	0	1

VFDT: Illustrative Evaluation – Error



VFDT: Illustrative Evaluation – Learning Time




Configure
EvaluatePrequential -l trees.HoeffdingTree -s generators.WaveformGenerator
Run
command
status
time elapsed
current activity
% complete
EvaluatePrequential -l (trees.Ho...
running
10m11s
Evaluating learner...
21,22
EvaluatePrequential -l trees.Ho...
running
11m13s
Evaluating learner...
12,25
Pause
Resume
Cancel
Delete
Preview (11m13s)
Refresh
Auto refresh:
every second
34982E-7,8900000.0,84.6,76.90361458431755,8900000.0,-11239.0,3211.0,1606.0,1606.0,24.0,0.0,0.0,-Infinity
34488E-7,9000000.0,83.8,75.6566322904254,9000000.0,-11330.0,3237.0,1619.0,1619.0,25.0,0.0,0.0,-Infinity
67947E-7,9100000.0,86.0,78.92784895482129,9100000.0,-11505.0,3287.0,1644.0,1644.0,25.0,0.0,0.0,-Infinity
17032E-7,9200000.0,86.1,79.14785222877956,9200000.0,-11589.0,3311.0,1656.0,1656.0,25.0,0.0,0.0,-Infinity
81544E-7,9300000.0,85.39999999999999,78.11360239611082,9300000.0,-11757.0,3359.0,1680.0,1680.0,25.0,0.0,0.0,-Infinity
92432E-7,9400000.0,85.0,77.47744365982531,9400000.0,-11841.0,3383.0,1692.0,1692.0,25.0,0.0,0.0,-Infinity
08526E-7,9500000.0,85.3,77.89839274706439,9500000.0,-11960.0,3417.0,1709.0,1709.0,25.0,0.0,0.0,-Infinity
92083E-7,9600000.0,84.89999999999999,77.34827846916366,9600000.0,-12100.0,3457.0,1729.0,1729.0,25.0,0.0,0.0,-Infinity
5001E-7,9700000.0,85.1,77.62678779233454,9700000.0,-12219.0,3491.0,1746.0,1746.0,25.0,0.0,0.0,-Infinity
Export as .txt file...
Evaluation
Values
Measure
Current
Mean
Accuracy 84,90 83,30 85,06 81,43
Kappa 77,30 74,91 77,57 72,13
Ram-Hours 0,00 0,00 0,00 0,00
Time 670,60 481,87 338,76 237,17
Memory 0,01 0,02 0,01 0,01
Plot
Zoom in Y
Zoom out Y
Zoom in X
Zoom out X
88,00

VFDT: Developments

- Regression:
  E. Ikonomovska, J. Gama, S. Dzeroski: *Learning model trees from evolving data streams*. Data Min. Knowl. Discov. 2011
- Rules:
  J. Gama, P. Kosina: *Learning Decision Rules from Data Streams*, IJCAI 2011
- Multiple Models:
  A. Bifet, E. Frank, G. Holmes, B. Pfahringer: Ensembles of Restricted Hoeffding Trees. ACM TIST; 2012
- . . .

Outline

Concept Drift

Incremental Decision Trees able to detect and react to concept drift

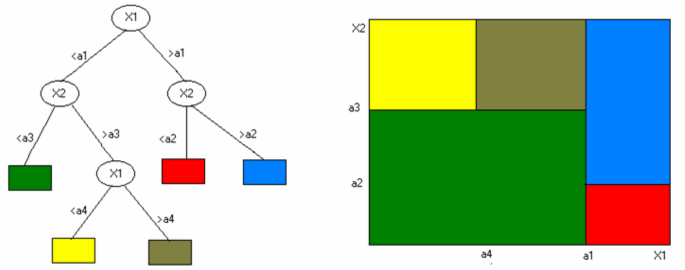
- Mining Time-Changing Data Streams
  - When a splitting-test is no more appropriate starts learning an alternate tree
  - G. Hulten, L. Spencer, P. Domingos; Kdd 2001
- Decision Trees for Dynamic Data Streams
  - Continuously monitors the error of a naive-Bayes in each node of a decision tree.
  - J. Gama, P. Medas, P. Rodrigues; SAC 2005
- Decision Trees for Mining Data Streams. IDA 10(1), 2006.
  - Compare the error distribution in two different time-windows;
  - J. Gama, R. Fernandes, R. Rocha:

Granularity of Decision Models

Occurrences of drift can have impact in part of the instance space.

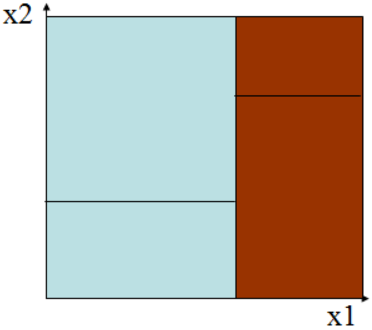
- **Global models:** Require the reconstruction of all the decision model. (like naive Bayes, SVM, etc)
- **Granular decision models**: Require the reconstruction of parts of the decision model (like decision rules, decision trees)

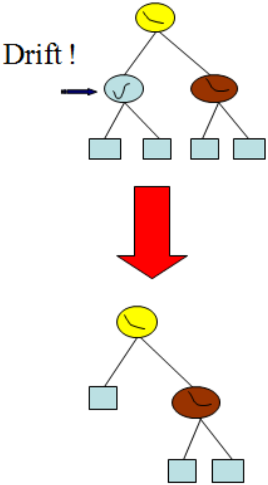
Detectors in each node!



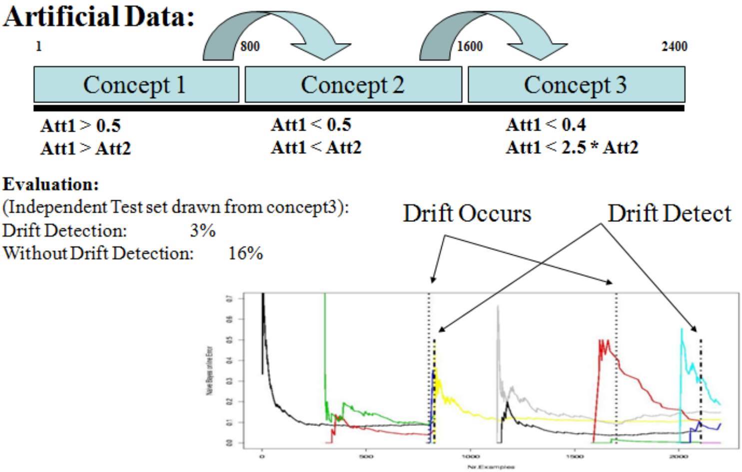
Detecting Drift

Each node has a naive-Bayes classifier, equipped with the SPC change detection algorithm.



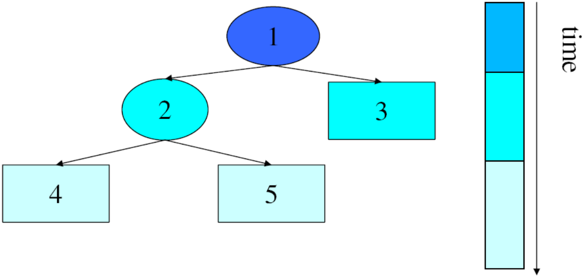


Concept Drift: Evaluation



VFDT like algorithms: Multi-Time-Windows

A multi-window system: each node (and leaves) receive examples from different time-windows.



Change detection based on distances between two time-windows.

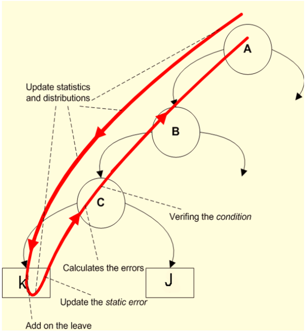
The RS Method

Implemented in the VFDTc system (IDA 2006)

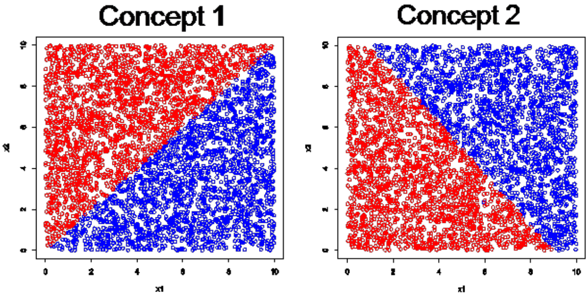
- For each decision node i, two estimates of the classification error.
  - Static error (SE_i): the distribution of the error of the node i;
  - Backed up error (BUE_i): the sum of the error distributions of all the descending leaves of the node i;
- With these two distributions:
  - we can detect the concept change,
  - by verifying the condition $SE_i \leq BUE_i$

The RS Method

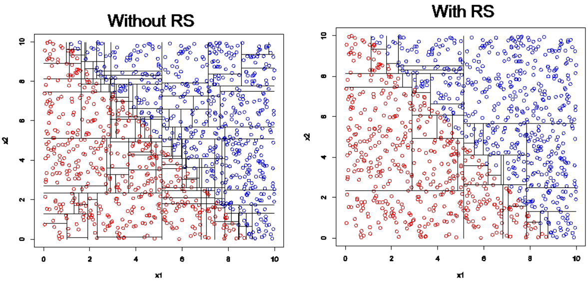
- Each new example traverses the tree from the root to a leaf
- Update the sufficient statistics and the class distributions of the nodes
- At the leaf update the value of SE_i
- It makes the opposite path, and update the values of SE_i and BUE_i for each decision node,
- Verify the regularization condition $SE_i \leq BUE_i$.
- If $SE_i \leq BUE_i$, then the node i is pruned to a new leaf.



The RS Method



The RS Method



Outline

VFDT: Analysis

The number of examples required to expand a node only depends on the Hoeffding bound.

- Low variance models:
  Stable decisions with statistical support.
- No need for pruning;
  Decisions with statistical support;
- Low overfiting:
  Examples are processed only once.
- **Convergence**: VFDT becomes asymptotically close to that of a batch learner. The expected disagreement is δ/p; where p is the probability that an example fall into a leaf.

Outline

Software

- VFML
  http://www.cs.washington.edu/dm/vfml/
  Very Fast Machine Learning toolkit for mining high-speed data streams and very large data sets.
- MOA
  http://sourceforge.net/projects/moa-datastream/
  A framework for learning from a data stream. Includes tools for evaluation and a collection of machine learning algorithms. Related to the WEKA project, also written in Java, while scaling to more demanding problems.
- Rapid Miner
  http://rapid-i.com/
  The Data Stream plugin provides operators for data stream mining and for learning drifting concepts.

Bibliography on Predictive Learning

- *Mining High Speed Data Streams*, by Domingos, Hulten, SIGKDD 2000.
- *Mining time-changing data streams*, Hulten, Spencer, Domingos, KDD 2001.
- *Efficient Decision Tree Construction on Streaming Data*, by R. Jin, G. Agrawal, SIGKDD 2003.
- *Accurate Decision Trees for Mining High Speed Data Streams*, by J. Gama, R. Rocha, P. Medas, SIGKDD 2003.
- *Forest trees for on-line data*; J. Gama, P. Medas, R. Rocha; SAC 2004.

Bibliography on Predictive Learning

- *Sequential inductive learning*, J. Gratch, AAAI, 1995.
- *Efficient Incremental Induction of decisions trees*, D. Kalles, Machine Learning, 1995
- *Learning decision trees from dynamic data streams*, Gama, Medas, and Rodrigues; SAC 2005
- *Decision trees for mining data streams*, Gama, Fernandes, and Rocha, Intelligent Data Analysis, Vol. 10, 2006.
- *Handling Time Changing Data with Adaptive Very Fast Decision Rules*, Kosina, Gama; ECML-PKDD 2012
- *Learning model trees from evolving data streams*, Ikonomovska, Gama, Dzeroski: Data Min. Knowl. Discov. 2011