Learning with Local Drift Detection

J.Gama, G.Castillo
LIACC-University of Porto

Outline

- Motivation
- Part I
  - Tracking Concept Drift
    - The Nature of Change
    - Detection Methods
    - Adaptation Methods
- Part II
  - Detecting Concept Drift
  - Local Drift Detection
    - Decision Trees for Data Streams
- Part III
  - Experimental Evaluation
- Conclusions

Motivation

- The Problem:
  - In most challenge applications of machine learning
    - Data flows continuously over time
    - Dynamic Environments
      - Some characteristic properties of the problem can change over time
  - Examples:
    - *Sensor Data*
    - *User modelling, e-commerce*
    - *Fraud Detection, Intrusion detection*
    - *Monitoring in biomedicine and industrial processes*
    - *Signal processing, Time series analysis, Automatic Control*
- Machine Learning algorithms assume:
  - instances are generated at random according to some stationary probability distribution *D*.
  - Instances are independent and identically distributed
  - It is required that D is stationary

Motivation

- Concepts are not static, can change over time
- Example:
  - **User Modeling Systems**:
    - to help users to find information
    - to recommend products
    - to adapt an interface, etc.

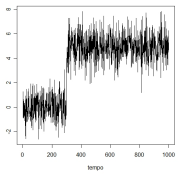
Can change over time:
•User's needs
•User's preferences
•Characteristics of the environment

- We are talking about learning systems
  - Learn from data streams in dynamic environments
    - instances are generated at random according to some **non-stationary** probability distribution
      - Presence of Hidden Contexts
  - **Evolve in Time: Incremental, Real Time,**
- **Learning in the presence of drift requires monitoring the learning process**

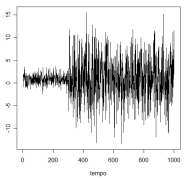
Concept Drift

- Concept drift means that the concept about which data is obtained may shift from time to time, each time after some minimum permanence.
  - Any change in the distribution underlying the data

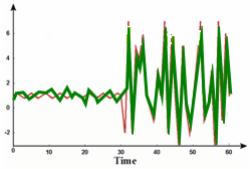
Change on the mean



Change on the variance



Change on the correlation



Tracking Concept Drift

The Nature of Change

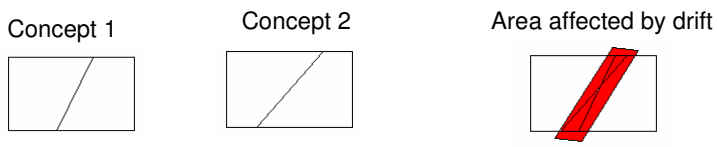
- The **causes** of change
  - Changes due to modifications in the context of learning due to changes in hidden variables
    - Important properties of the domain that are not observed.
    - Example:
      - People buy different objects in winter and summer.
      - Butterflies in Beijing and storms in Portugal
  - Changes in the characteristic properties of the observed variables.
    - Example:
      - Small faults in parts of an industrial process can change the quality of the product

The Nature of Change

- The **rate** of change:
  - Concept Drift
    - Gradual changes in the target concept
  - Concept Shift
    - Refers to abrupt changes.
- Often crucial is detection of gradual modifications which slowly affect the learning process.
- Different from the amplitude of change

The Nature of Change

- Whenever a change in the underlying concept generating data occurs, the class-distribution of examples changes.
  - At least in some regions of the instance space.
    - Wei Fan. *Systematic data selection to mine concept-drifting data streams.* Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining. ACM Press, 2004.
- It is possible to observe changes in the class-distribution without concept drift.
  - This is usually referred as **virtual drift**



Algorithms for Change Detection

- Goals of change detection algorithms:
  - Few false alarms and missed detections
  - Low detection delay
- Change Detection Algorithms:
  - Statistical Process Control (Quality Control)
    - Control Charts
      - Shewhart Charts, p-charts, …
  - Stochastic Process, Time Series
    - Recursive Cumulative Sums (CUSUM)
    - Sequential Probability Ratio Test
    - Generalized log-likelihood Ratio
- Methodologies
  - Sequential Statistics
    - A.P. Dawid, 1984

Algorithms for Change Detection

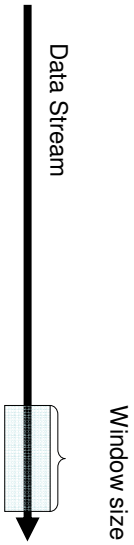
- Key IDEA: Forgetting old information.
- We need Incremental and Decremental Learning Algorithms!
  - Algorithms that can incorporate new information and forget oldest information.

Characterization of Drift Detection Methods

- Data management
  - Characterizes the information about training examples stored in memory.
- Detection methods
  - Characterizes the techniques and mechanisms for drift detection
- Adaptation methods
  - Adaptation of the decision model to the actual distribution
- Decision model management

Data Management

- Characterize the information stored in memory to maintain a decision model consistent with the actual state of the nature.
- Full Memory.
  - Store in memory sufficient statistics over all the examples.
    - Weighting the examples accordingly to their age.
    - Oldest examples are less important.
- Partial Memory.
  - Store in memory only the most recent examples.
  - Examples are stored in a *first-in first-out* data structure.
    - A *window* of the most recent examples
    - At each time step the learner induces a decision model using only the examples that are included in the window.



Data Management

- Partial Memory.
  - The key difficulty is how to select the appropriate window size:
    - small window
      - Can assure a fast adaptability in phases with concept changes
      - In more stable phases it can affect the learner performance
    - large window
      - Produce good and stable learning results in stable phases
      - Can not react quickly to concept changes.

Data Management

- Partial Memory.
  - Fixed Size windows.
    - Store in memory a fixed number of the most recent examples.
    - Whenever a new example is available:
      - » it is stored in memory and the oldest one is discarded.
    - This is the simplest method to deal with concept drift and
      - » Used as a baseline for comparisons.
  - Adaptive Size windows.
    - the set of examples in the window is variable.
    - They are used in conjunction with a detection model.
    - Decreasing the size of the window whenever the detection model signals drift and increasing otherwise.

Data Management

- The data management model also indicates the forgetting mechanism.
  - Weighting examples corresponds to a gradual forgetting.
  - The relevance of old information is less and less important.
  - Time windows corresponds to abrupt forgetting.
    - Oldest examples are deleted from memory.
- Of course we can combine both forgetting mechanisms by weighting the examples in a time window.

Detection Methods

- The Detection Model characterizes the techniques and mechanisms for drift detection.
- An advantage of the detection model is they can provide:
  - meaningful description about evolution of data
    - indicating change-points or
    - small time-windows where the change occurs
  - quantification of the changes.

Detection Methods

- Monitoring the evolution of performance indicators.
  - Some indicators are monitored over time
    - Performance measures,
    - Properties of the data, etc
- Monitoring distributions on two different time-windows.

Monitoring the Evolution of Performance Indicators

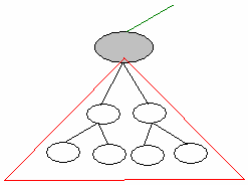
- The FLORA family of algorithms
  - Developed by Widmer and Kubat (1996).
  - FLORA2 includes a window adjustment heuristic for a rule-based classifier.
  - To detect concept changes the **accuracy** and the **coverage** of the current learner are monitored over time and the window size is adapted accordingly.
  - FLORA3: dealing with recurring concepts.
- Theoretical Leave-one-out estimators for SVM
  - **Accuracy**, **recall** and **precision** over time, and then, comparing it to a confidence interval of standard sample errors for a moving average value using the last *M* batches of each particular indicator.
    - Developed by Klinkenberg (98)

Monitoring distributions on two different time-windows

- D.Kifer, S.Ben-David, J.Gehrke (VLDB 04)
  - Propose algorithms (statistical tests based on Chernoff bound) that examine samples drawn from two probability distributions and decide whether these distributions are different.
- Gama, Fernandes, & Rocha: *VFDTc* (IDA 2006)
  - Continuously monitoring differences between two class-distribution of the examples:
    - the distribution when a node was built and the class-distribution when a node was a leaf and
    - the weighted sum of the class-distributions in the leaves descendent of that node.

The Sequential Regularization (SR) Method

- Implemented in the VFDTc system
- 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 $_i$



Adaptation Methods

- The Adaptation model characterizes the changes in the decision model to adapt to the most recent examples.
- Blind Methods:
  - Methods that adapt the learner at regular intervals without considering whether changes have really occurred.
    - Examples include methods that weight the examples accordingly to their age and methods that use time-windows of fixed size.
- Informed Methods:
  - Methods that only change the decision model after a change was detected. They are used in conjunction with a detection model.
- Adaptation Methods depends on the model-class used to generalize examples.

Decision Model Management

- Model management characterize the number of decision models needed to maintain in memory.
- Using Multiple Models:
  - The key issue here is the assumption that data generated comes from multiple distributions,
    - at least in the transition between contexts.
  - Instead of maintaining a single decision model several authors propose the use of multiple decision models.
  - Seasonality: re-occurring concepts
    - G. Widmer and M. Kubat; *Learning in the presence of concept drift and hidden contexts*. Machine Learning, 23:69--101, 1996(FLORA3)
    - Ying Yang, Xindong Wu, and Xingquan Zhu Mining in Anticipation for Concept Change: Proactive-Reactive Prediction in Data Streams, *Data Mining and Knowledge Discovery*, (to appear)

Decision Model Management

- A seminal work, is the system presented by Kolter and Maloof (ICDM03, ICML05).
  - The Dynamic Weighted Majority algorithm (DWM) is an ensemble method for tracking concept drift.
  - Maintains an ensemble of base learners,
    - Predicts using a weighted-majority vote of these ”experts”.
    - Dynamically creates and deletes experts in response to changes in performance.

Decision Model Management

- Granularity of Decision Models
  - Occurences 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)

Detecting Drift

Detecting Drift

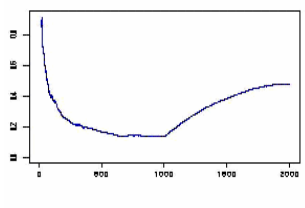
- The Basic Idea:
  - When there is a change in the class-distribution of the examples:
    - The actual model does not correspond any more to the actual distribution.
    - The error-rate increases
- Main Issues:
  - Detect when the actual model is out-date
    - Trace of the error rate
  - React to drift
    - Re-learn the decision model using the most recent examples
      - Dynamic Window
        - » *Short Term Memory*

Detecting Drift

- Suppose a sequence of examples in the form $<x_i,y_i>$
- The actual decision model classifies each example in the sequence
  - In the 0-1 loss function, predictions are either True or False
    - The predictions of the learning algorithm are: T,F,T,F,T,F,T,T,T,F,….
    - A random variable from Bernoulli trials
- The Binomial distribution gives the general form of the probability of observing a F
    - $p_i = (\#F/i)$
    - $S_i = \sqrt{p_i(1-p_i)/i}$
      - *Where i is the number of trials*

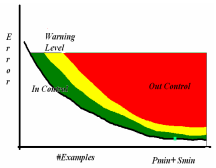
Monitoring the Error Rate

- We define *context* as a sequence of examples where
  - The function governing the distribution of examples is stationary.
  - In a context, the error rate
    - Could decrease
    - Should not significantly increase.
- Monitoring the trace of the p_i and s_i
- Concept Drift occurs:
  - When the error significantly increases
    - Change in the context
- A datastream is a sequence of contexts



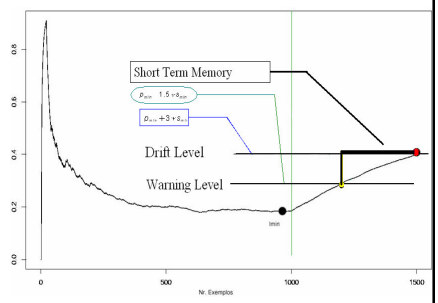
Detect Drift

- The algorithm maintains two registers
  - P_{min} and S_{min} such that $P_{min}+S_{min} = \min(p_i+s_i)$
    - Minimum of the Error rate taking the variance of the estimator into account.
- At example *j*
  - The error of the learning algorithm will be
    - **Out-control** if $p_j+s_j > p_{min}+ \alpha * s_{min}$
    - **In-control** if $p_j+s_j < p_{min}+ \beta * s_{min}$
    - Warning if $p_{min}+ \alpha * s_{min} > p_j+s_j > p_{min}+ \beta * s_{min}$
      - » The constants α and β depend on the confidence level
      - » In our experiments β=2 and α = 3



The Algorithm

- At example *j* the actual model classifies the example
  - Compute the error and variance: p_j and s_j
  - If the error is
    - In-control the actual model is updated
      - Incorporate the example in the decision model
    - Warning zone:
      - Maintain the actual model
      - First Time:
        - » the lower limit of the window is: $L_{warning} = j$
    - Out-Control
      - Re-learn a new model using as training set the set of examples $[L_{warning}, j]$

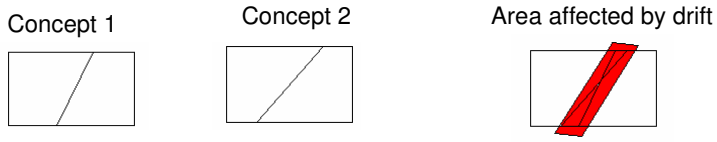


Data streams: sequences of contexts

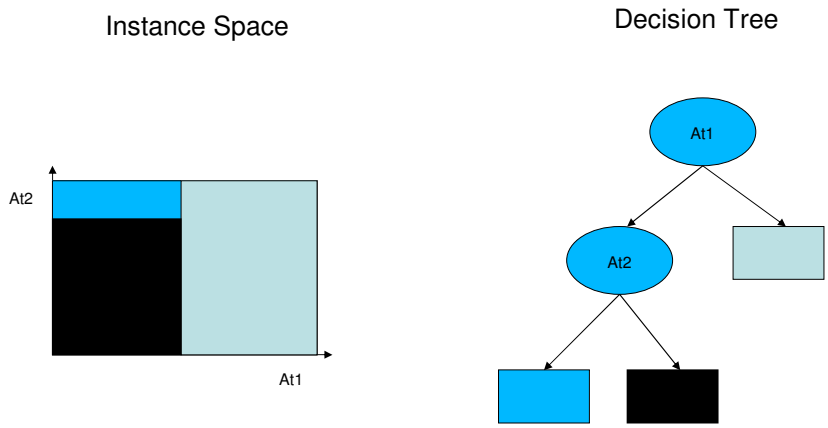
- The Drift Detection Method can be used as a wrapper over a learning algorithm
- Whenever a new context is detect
  - The actual decision is forgotten
  - A new decision model is learn
    - Using the examples is the short term memory.
      - The most recent examples
- The process of monitoring the error is re-initialized
- When Drift is Detected a new decision model is learn from scratch
  - Problem: All the decision Model must be learn from scratch!
  - Only useful if the concept change in the whole instance space.

Local Drift Detection

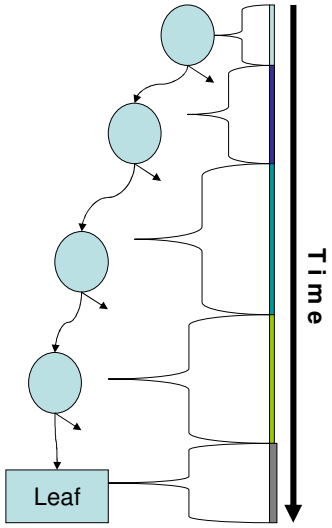
- Occurrence of drift can affect part of the instance space
- Granular Decision Models
  - Fit different models to different regions of the instance space
    - Rule learners, Decision Trees
  - The adaptation phase can be restricted to parts of the decision model.
  - Avoid expensive update of the global model.
-



Granular Decision Models



VFDT like Decision Tree Algorithms

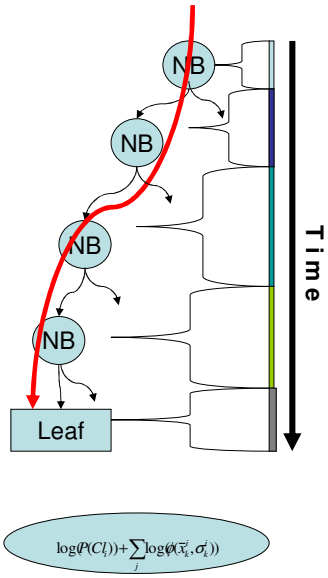


- Learning from data streams
- The decision to expand a leaf
  - Is based only in examples from a time window
  - Requires statistical Support (Hoeffding bound)
- Nodes receive information from different time-windows
  - Information about the most recent examples is stored in the leaves
  - Past Information stored in inner-nodes

Ultra Fast Forest of Trees

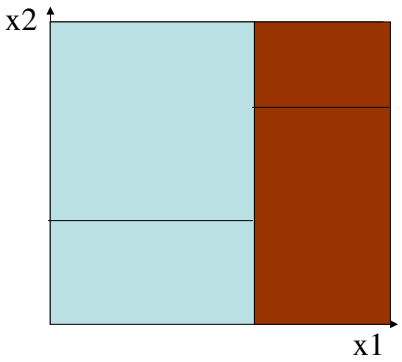
- UFFT: a VFDT like algorithm
  - Incremental, online forest of trees for data-streams
  - Processes each example in constant time and memory
  - Single scan over the data
  - Fast Splitting Criteria:
    - Based on quadratic discriminant analysis
    - The sufficient statistics:
      - Mean and standard deviation per attribute per class
      - Continuous attributes
  - From the sufficient statistics is immediate to derive:
    - Naïve Bayes classifiers
      - Functional Leaves
        - » Anytime Classifier
      - **Drift Detectors**

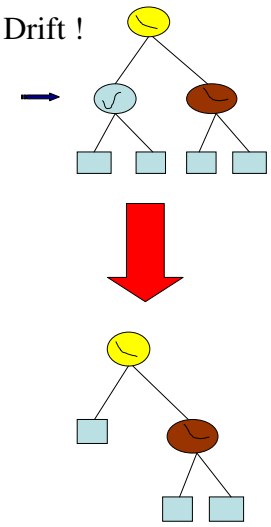
Drift Detection in Ultra Fast Forest of Trees



- Each node in the tree maintains a naive-Bayes classifier
- Whenever a training example is available
  - It is classified by each naïve-Bayes installed at inner-nodes.
  - Drift Detection:
    - Monitor the naïve-Bayes error
    - Signal Drift
      - When the error is out-of-control
    - Model Adaptation
      - Pruning the subtree rooted at that node.

Local Drift Detection





Experimental Evaluation

Methodological Issues

- Evaluation under drift: what should we evaluate?
  - Performance Indexes:
    - Probability of False alarms
    - Probability of non-detection
    - Mean delay for detection
  - How to Evaluate?
    - Resilience to False Alarms when there is no drift.
      - Stationary environments
    - Capacity to Detect and React to drift.
      - Dynamic environments
    - Evolution of Decision model Error
      - During the learning process.

Methodological Issues

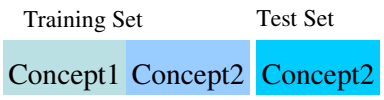
- What should we evaluate?
  - Different types of drift
    - Fast/Slow changes
  - Two types of Errors
    - Type I error: False Alarms
      - Detecting drift when drift did not occur
    - Type II error: True changes not detected
      - Not detecting drift when it occurs
  - Fast Detection
    - How many examples required to detect occurrence of drift

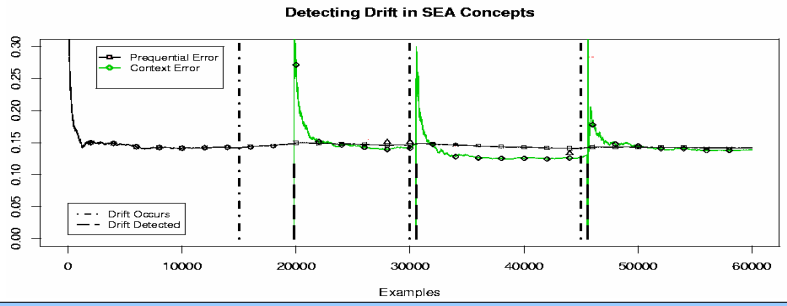
Methodological Issues

- What should we evaluate?
  - Ability to Detect drift.
  - Resilience to False alarms
  - Fast detection Rate
- How to measure ?
  - Controlled experiments with artificial data
    - Randomly generate sets of examples for each concept
    - Training sets are composed by sequences of concepts
    - Evaluation of the resulting models:
      - In a test set using the last concept
      - **Prequential Error:**
        - » **Sequence of predictions for the next example**

Methodological Issues

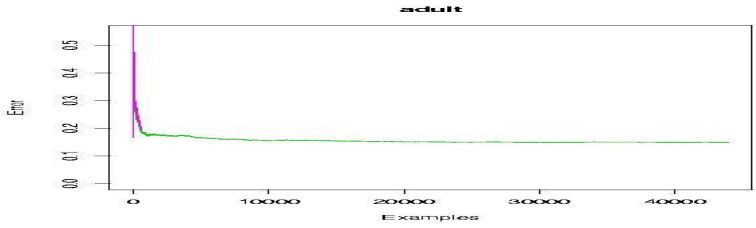
- Decision Model Error
  - Error on independent test set.
  - Evolution of *prequential* error
    - **Sequence of predictions for the next example**
    - For each example
      - First: Predict Target Label
      - Next: Update Decision Model





Stationary Data

- *Adult* UCI Data Set
  - The algorithm didn't detect
    - Nor drift
    - Nor warning
- Resilience to False Positives



Evaluation in Wrapper Mode

Artificial Data

- 8 artificial data previously used in drift detection (Kubat, et. al 1995)
- All problems are two class problems.
- Random examples
- Problems with 2 or more concepts
- Each concept contains 1000 examples

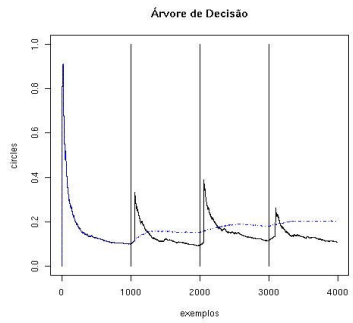
- Main Characteristics:
  - Type of change
    - Gradual
    - Abrupt
  - Noise
    - With/Without noise
  - Data Types
    - Homogeneous
    - Heterogeneous
  - Irrelevant Attributes
    - Yes/No

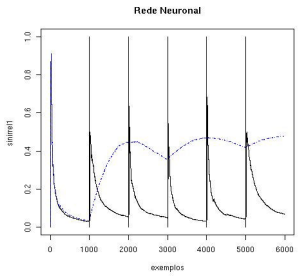
Learning Algorithms

- The drift detection method acts as a wrapper over batch of the algorithms:
  - Perceptron
    - Linear decision surface
  - Neural Network
    - Non-linear decision surface
  - Decision Trees
    - Decision surface defined by hyper-rectangles
      - Disjunctive normal form

Artificial Data: Results

- Problem *Circles*
  - Nr. of Concepts: 4
    - Gradual change of the concept
    - Without noise
- Problem: *Sinirrel*
  - Nr. of Concepts: 2x2x2
    - Abrupt Change
    - 2 Irrelevanr Attributes
    - Noisy





Results of Drift Detection

Data Set	Perceptron		Neural Net		Decision Tree	
	Without	With	without	with	without	with
STAGGER	0,048	**0,029**	0,351	**0,002**	0,265	**0,016**
Sine1	0,126	**0,115**	0,489	**0,019**	0,490	**0,081**
SINIrrel1	0,159	**0,139**	0,479	**0,068**	0,483	**0,088**
Sine2	0,271	**0,262**	0,492	**0,118**	0,477	**0,100**
SINIrrel2	0,281	0,281	0,477	**0,059**	0,485	**0,084**
Mixed	0,100	0,111	0,240	**0,065**	0,491	**0,465**
Gauss	0,384	0,386	0,395	**0,150**	0,380	**0,144**
Circle	0,410	0,413	0,233	**0,225**	0,205	**0,109**

Drift Evaluation

Artificial Data:

1	800	1600	2400
Concept 1	Concept 2	Concept 3	
Att1 > 0.5 **Att1 > Att2**	**Att1 < 0.5** **Att1 < Att2**	**Att1 < 0.4** **Att1 < 2.5 * Att2**	

Evaluation:

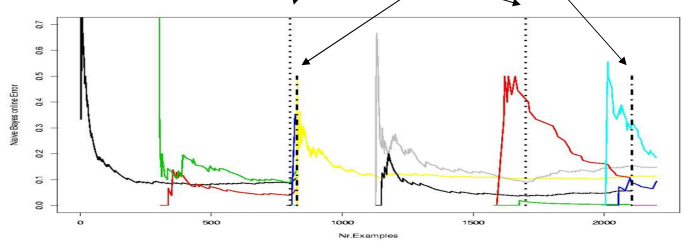
(Independent Test set drawn from concept3):

Drift Detection: **3%**

Without Drift Detection: **16%**

Drift Occurs

Drift Detect

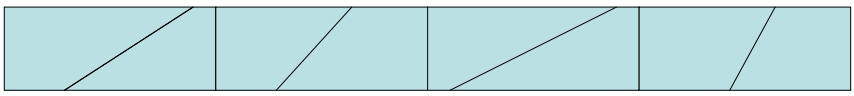


SEA Concepts

- N. Street and Y. Kim, *A streaming ensemble algorithm (SEA) for large-scale classification*; KDD 01
- Data Generation
  - Generate 50000 random points in a three-dimensional space.
  - Those points are divided into 4 blocks with different concepts.
  - Hyper-plan Decision Surface: $f1+f2 < \Theta$
    - Threshold values of 8, 9, 7 and 9.5
  - Class noise in the class label 10%
    - Note that attribute f_3 is irrelevant.

Amplitude of Change:

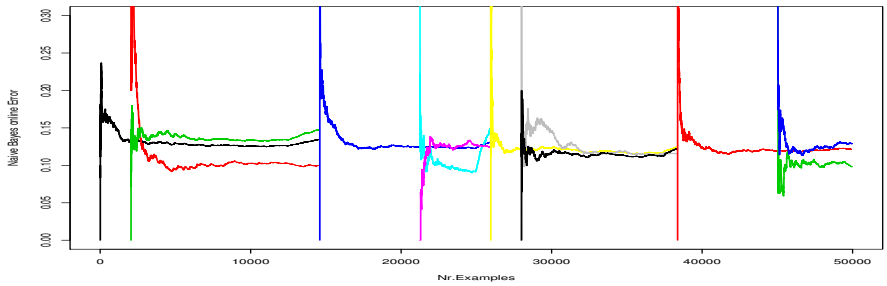
8	9	7	9.5



SEA Concepts

Streaming Ensemble Algorithm for large scale classification, N. Street,Y.Kim KDD01

	UFFT	UFFT-ND	CVFDT	VFDT	VFDTc
Error	12.99	15.89	14.72	16.06	14.40
t.test	-	0	0.002	0	0.0001



The Electricity Market Dataset

- The data was collected from the Australian New South Wales Electricity Market
  - The electricity price is not fixed
    - The price is set every 5 minutes
    - It is affected by demand and supply of the market
  - The dataset covers the period from 7 May 1996 till 5 December 1998
    - Contains 45312 examples
    - Attributes
      - Day of Week
      - NSW electricity demand
      - Victorian electricity demand
      - Scheduled electricity transfer
      - …
      - Class Label:
        - » Change (UP, DOWN) of the price related to a moving average of the last 24 hours.

Experiments

- Two sets of experiments:
  - Predicting last week
    - Training set: 35482 examples
  - Predicting last day
    - Training set: 38362 examples
- Error-rates using the decision tree available in R (CART like):

Test Set	All Data	Last Year
Last Day	18.7%	12.5%
Last Week	23.5%	22.4%

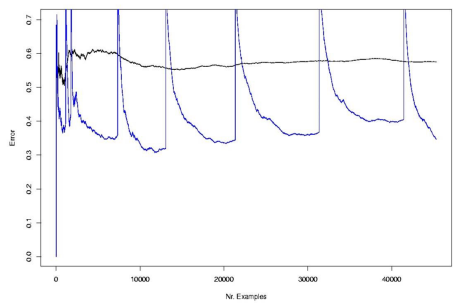
Results using Drift Detection

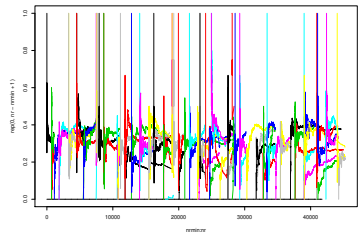
Test Set	All Data	Last Year	Drift Detection
Last Day	18.7%	12.5%	10.4%
Last Week	23.5%	22.4%	19.9%

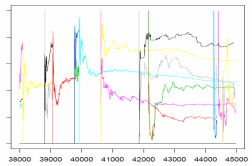
Online error

Trace of the online error of a decision tree:

- Using drift detection
- Without using drift detection



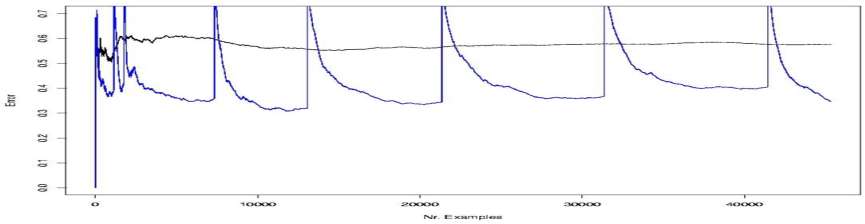




Online error

Trace of the online error of a decision tree:

- Using drift detection
- Without using drift detection



Test Set	Lower Bound	All Data	Last Year	Drift Detection
Last Day	10.4%	18.7%	12.5%	10.4%
Last Week	19.0%	23.5%	22.4%	19.9%

Conclusions

- Real-world applications of Machine Learning:
  - Data flows sequentially over time
  - Dynamic environments
    - Characteristic properties of data can change over time
- Change detection algorithms must be embedded into Learning Systems.
  - Resilient to False alarms
  - Effective in change detection
  - Low detection delay
- Insights about data evolution.

Conclusions: Change Detection Issues

- Data management
  - Characterizes the information about training examples stored in memory.
- Detection methods
  - Characterizes the techniques and mechanisms for drift detection
- Adaptation methods
  - Adaptation of the decision model to the actual distribution
- Decision model management

Conclusions

- Monitoring The Learning Process
  - Main Characteristics:
    - Resilient to False alarms
      - Low probability in signalling drift when it doesn't occurs
    - Effective in drift detection
      - High probability to detect drift when it occurs
    - Computationally efficient
    - Insights about data evolution.
      - Information about when drift occurs
- Statistical Process control
  - Tools for Monitoring the learn process

Conclusions

- Applicable both:
  - As a wrapper over a learning algorithm.
  - Embedded inside a granular decision model.
- Advantages of Local Drift Detection:
  - Fast in drift detection.
  - Efficient in model adaptation.
    - Changes affect part of decision model

Future Work

- Apply change detection mechanisms to:
  - Other loss functions
    - Mean squared error
  - Other learning problems
    - Clustering
    - Association rules
    - Summarization
- Real-world applications.

Bibliography

- Michele Basseville and Igor Nikiforov. *Detection of Abrupt Changes: Theory and Applications*, Prentice-Hall Inc, 1987.
- A.P. Dawid. *Statistical theory: The prequential approach*. Journal of the Royal Statistical Society-A, 147:278--292, 1984.
- Wei Fan. *Systematic data selection to mine concept-drifting data streams*. Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining. ACM Press, 2004.
- J.Gama, R.Fernandes, and R.Rocha. *Decision trees for mining data streams*. Intelligent Data Analysis, 10(1):23--46, 2006.
- João Gama, Pedro Medas, and Pedro Rodrigues. *Learning decision trees from dynamic data streams*. Proceedings of the 2005 ACM Symposium on Applied Computing, pages 573--577. ACM Press, March 2005.
- M.Harries, C.Sammut, and K.Horn. *Extracting hidden context*. Machine Learning, 32:101, 1998.
- Michael Harries. *Splice-2 comparative evaluation: Electricity pricing*. Technical report, The University of South Wales, 1999.

Bibliography

- Geoff Hulten, Laurie Spencer, and Pedro Domingos. *Mining time-changing data streams*. Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pages 97--106. ACM Press, 2001.
- Daniel Kifer, Shai Ben-David, and Johanes Gehrke. *Detecting change in data streams*. VLDB 04: Proceedings of the 30th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc., 2004.
- R.Klinkenberg. *Learning drifting concepts: Example selection vs. example weighting*. Intelligent Data Analysis, 2004.
- R.Klinkenberg and T.Joachims. *Detecting concept drift with support vector machines*. Proceedings of ICML-00, 17th International Conference on Machine Learning, pages 487--494, 2000. Morgan Kaufmann Publishers.
- R.Klinkenberg and I.Renz. *Adaptive information filtering: Learning in the presence of concept drifts*. In Learning for Text Categorization, pages 33--40. AAAI Press, 1998.
- J. Kolter and M Maloof. *Dynamic weighted majority: A new ensemble method for tracking concept drift*. In Proceedings of the Third International IEEE Conference on Data Mining, pages 123--130. IEEE Computer Society, 2003.
- J. Kolter and M.Maloof. *Using additive expert ensembles to cope with concept drift*. Machine Learning, Proceedings of the 22th International Conference. OmniPress, 2005.

Bibliography

- I.Koychev. *Gradual forgetting for adaptation to concept drift*. Proceedings of ECAI 2000 Workshop Current Issues in Spatio-Temporal Reasoning. Berlin, Germany, pages 101--106, 2000.
- I.Koychev. *Learning about user in the presence of hidden context*. Proceedings of Machine Learning for User Modeling: UM-2001, 2001.
- C. Lanquillon. *Enhancing Text Classification to Improve Information Filtering*. PhD thesis, University of Madgdeburg, Germany, 2001.
- M.Maloof and R.Michalski. *Selecting examples for partial memory learning*. Machine Learning, 41:27--52, 2000.
- N. Street and Y. Kim. *A streaming ensemble algorithm SEA for large-scale classification*. Proc. seventh ACM SIGKDD ACM Press, 2001.
- G. Widmer and M. Kubat; *Learning in the presence of concept drift and hidden contexts*. Machine Learning, 23:69--101, 1996.