ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

MDAKM
Multidimensional Data Analysis and Knowledge Management Laboratory®

Streaming Data Analysis and Management

Prof. Vasileios Megalooikonomou

Dept. of Computer Engineering & Informatics
University of Patras
Greece

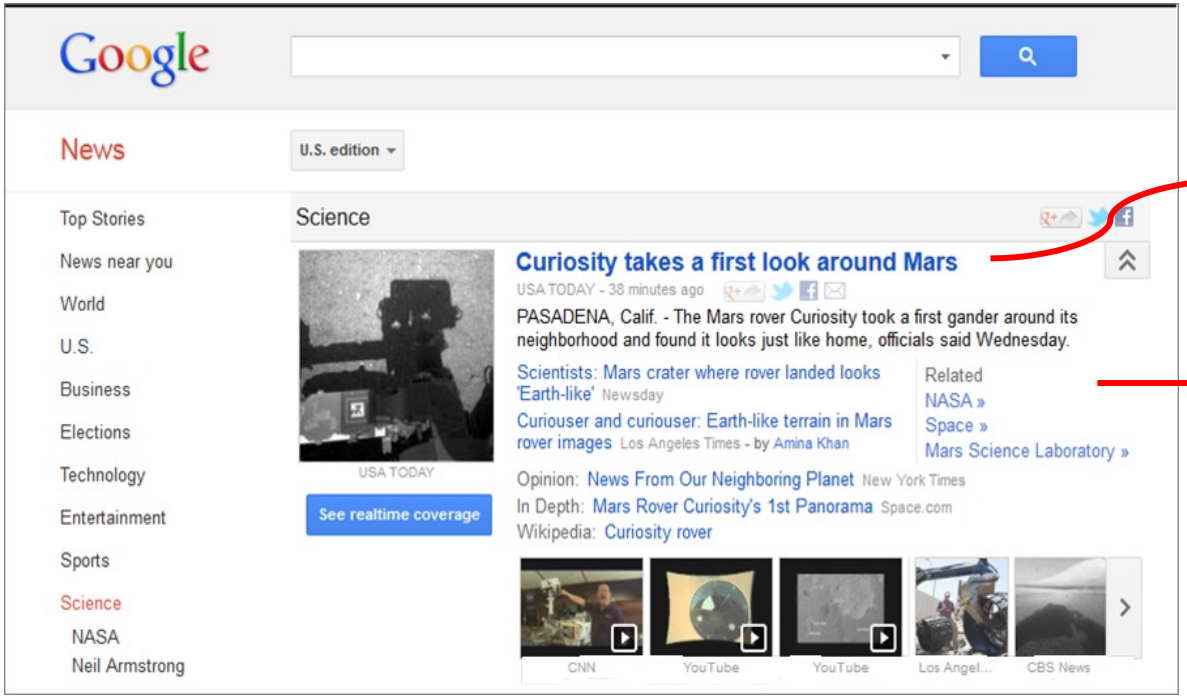
Stream analysis

- Stream: Continuous flow of data
- Challenges:
  - **Volume**: Not possible to store all the data
  - **One-time access**: Not possible to process the data using multiple passes
  - **Real-time analysis**: Certain applications need real-time analysis of the data
  - **Temporal Locality**: Data evolves over time, so model should be adaptive.

- Stream mining algorithms (I will not talk about)
  - How many distinct elements appear in a stream- provide an estimate (Flajolet-Martin)
  - Estimate the number of 1’ in a window (DGIM)
  - Estimate frequency moments (AMS)
  - Finding the most popular elements in the stream (Decaying windows – assign more weight to newer elements)
  - Identify of an element’s presence in a set (Bloom filters)

Clustering of streaming data

Stream Clustering



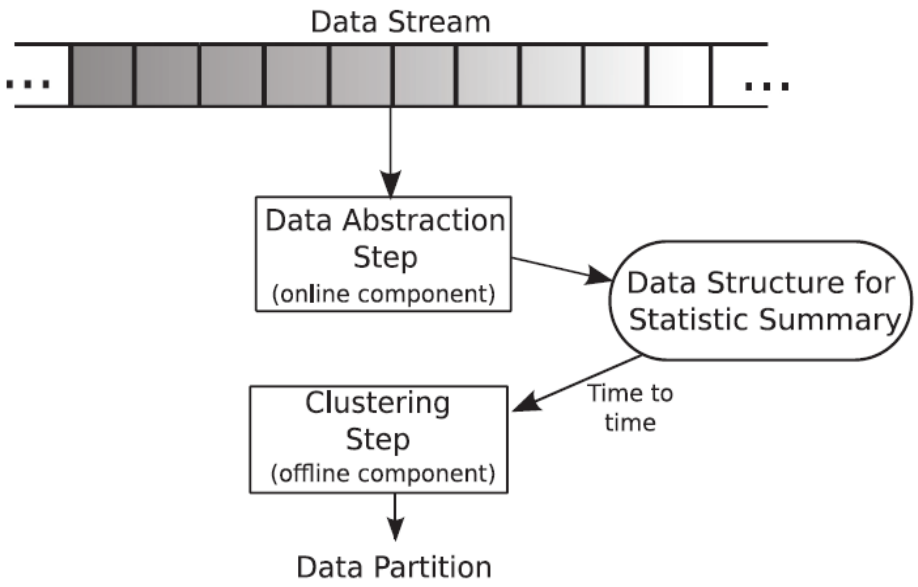
Stream Clustering

The problem of data stream clustering is defined as:

Input: a sequence of *n* points in metric space and an integer *k*.

Output: *k* centers in the set of the *n* points so as to minimize the sum of distances from data points to their closest cluster centers.

Stream Clustering



- Online Phase
  - Summarize the data into memory-efficient data structures
- Offline Phase
  - Use a clustering algorithm to find the data partition

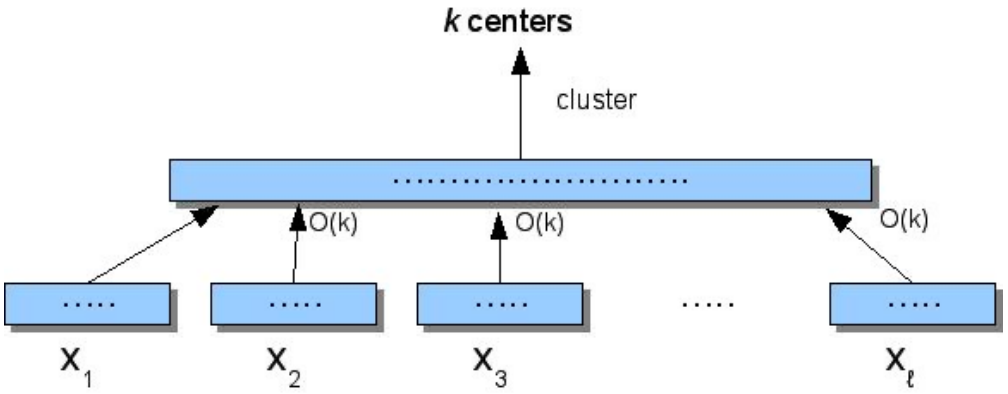
Stream Clustering Algorithms

Data Structures	Examples
Prototypes	Stream, Stream Lsearch
CF-Trees	Scalable k-means, single pass k-means
Microcluster Trees	ClusTree, DenStream, HP-Stream
Grids	D-Stream, ODAC
Coreset Tree	StreamKM++

Prototypes: STREAM

- Guha, Mishra, Motwani and O'Callaghan (2000)
- Achieves a constant factor approximation for the k-Median problem in a single pass and using small space.
- Small-Space:
  - a divide-and-conquer algorithm that divides the data, S, into l pieces, clusters each one of them (using k-means) and then clusters the centers obtained.
- Algorithm Small-Space (S):

  1. Divide S into l disjoint pieces $X_1,\ldots,X_l$
  2. For each i, find O(k) centers in X_i. Assign each point in X_i to its closest center.
  3. Let X' be the O(lk) centers obtained in (2), where each center c is weighted by the number of points assigned to it.
  4. Cluster X' to find k centers.



STREAM

- Problem with Small-Space:
  - number of subsets l is limited, since it has to store in memory the intermediate medians in *X*.
  - If M is the size of memory we need to partition *S* into l subsets such that each subset fits in memory, (n/ l) and so that the weighted lk centers also fit in memory, $lk<M$.
  - But such an l may not always exist!

- STREAM algorithm solves the problem of storing intermediate medians and achieves better running time and space requirements.

STREAM

1. Input the first *m* points; using the randomized algorithm reduce these to O(k) (say 2*k*) points.

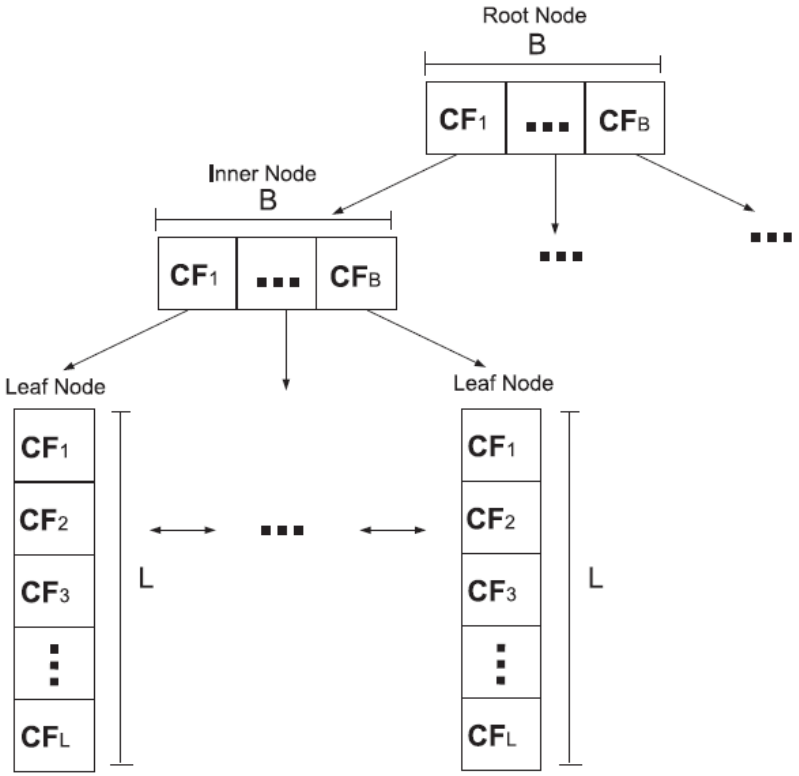
2. Repeat the above till we have seen $m^2/(2k)$ of the original data points. We now have *m* intermediate medians.

3. Using a local search algorithm, cluster these *m* first-level medians into 2*k* second-level medians and proceed.

4. In general, maintain at most *m* level-*i* medians, and, on seeing *m*, generate 2*k* level-*i*+ 1 medians, with the weight of a new median as the sum of the weights of the intermediate medians assigned to it.

5. When we have seen all the original data points, we cluster all the intermediate medians into *k* final medians, using the primal dual algorithm

CF-Trees



Summarize the data in each CF-vector

- N: Number of points
- LS: Linear sum of data points
- SS: Squared sum of data points

BIRCH, Scalable k-means, Single pass k-means

BIRCH

- BIRCH: Balanced Iterative Reducing and Clustering using Hierarchies, by Zhang, Ramakrishnan, Livny (SIGMOD'96)
- Received the SIGMOD 10 year test of time award in 2006
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
  - **Phase 1**: scan DB to build an initial in-memory CF tree
    - a multi-level compression of the data that tries to preserve the inherent clustering structure of the data
  - **Phase 2**: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree

BIRCH

- *Strengths:*
  - *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
  - *It is local* in that each clustering decision is made without scanning all data points and currently existing clusters.
  - It exploits the fact that *data space is not usually uniformly occupied and not every data point is equally important*.
  - It makes full use of available memory to derive the finest possible sub-clusters while *minimizing I/O costs*.
  - It is also an *incremental method* that does not require the whole data set in advance
- *Weakness:* handles only numeric data and is sensitive to the order of the data record.

Clustering Feature Vector

Clustering Feature: $CF = (N, \overrightarrow{LS}, SS)$

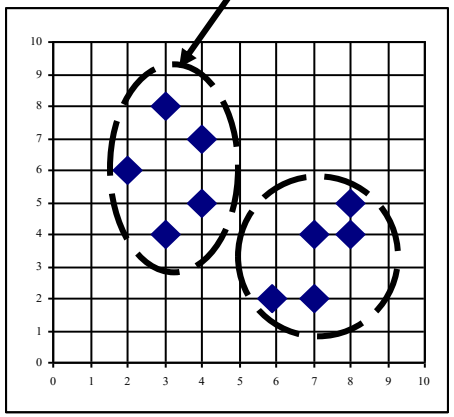
N: **Number of data points**

$\overrightarrow{LS}$: $\sum_{i=1}^{N} \overrightarrow{X_i}$

SS: $\sum_{i=1}^{N} (\overrightarrow{X_i})^2$

Clustering features are additive

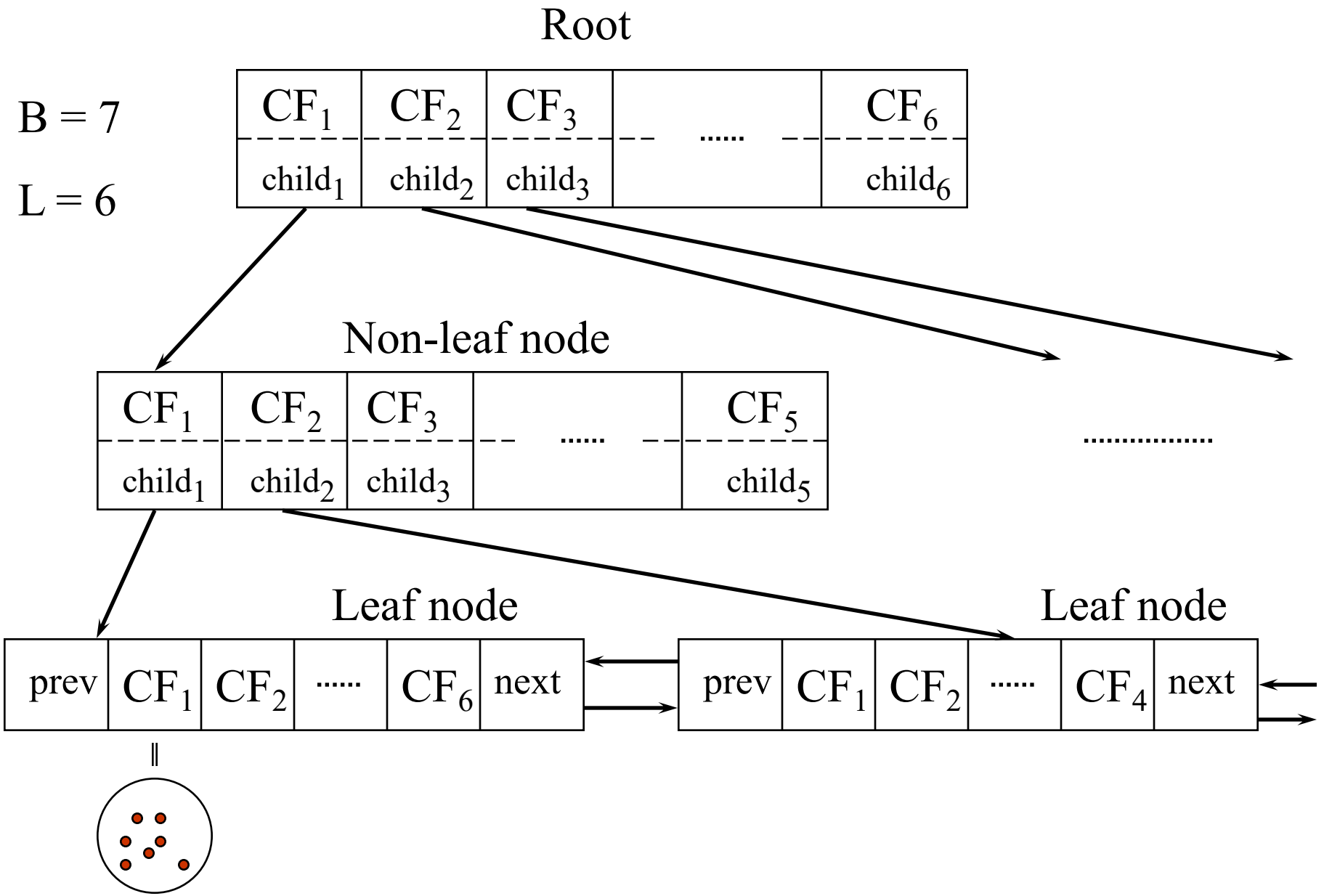
CF = (5, (16,30),(54,190))



(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

CF Tree

B: **Branching factor**: max # children per nonleaf node
L: **Threshold**: max diameter of subclusters at leaf nodes



Calculations

Given CF = (N, LS, SS) the same measures can be calculated without the knowledge of the underlying actual values

- Centroid: $\overrightarrow{C} = \frac{\sum_{i=1}^{N} \overrightarrow{X_i}}{N} = \frac{\overrightarrow{LS}}{N}$
- Radius: $R = \sqrt{\frac{\sum_{i=1}^{N} (\overrightarrow{X_i} - \overrightarrow{C})^2}{N}} = \sqrt{\frac{N \cdot \overrightarrow{C}^2 + SS - 2 \cdot \overrightarrow{C} \cdot \overrightarrow{LS}}{N}} = \sqrt{\frac{SS}{N} - (\frac{\overrightarrow{LS}}{N})^2}$
- Average Linkage Distance between clusters $CF_1 = [N_1, \overrightarrow{LS_1}, SS_1]$ and $CF_2 = [N_2, \overrightarrow{LS_2}, SS_2]$:

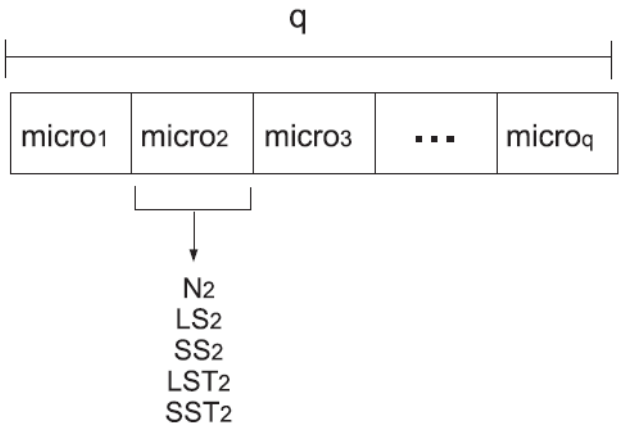
$$D_2 = \sqrt{\frac{\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (\overrightarrow{X_i} - \overrightarrow{Y_j})^2}{N_1 \cdot N_2}} = \sqrt{\frac{N_1 \cdot SS_2 + N_2 \cdot SS_1 - 2 \cdot \overrightarrow{LS_1} \cdot \overrightarrow{LS_2}}{N_1 \cdot N_2}}$$

Subtracting good approximations to two nearby numbers may yield a very bad approximation to the difference of the original numbers -> Catastrophic cancellation

Use **BETULA cluster features *(N, μ, S)*** instead where *N* is the count, *μ* the mean and *S* the sum of squared deviations (based on numerically more reliable online algorithms to calculate variance).

Microclusters

CF-Trees with “time” element



CluStream

- Linear sum and square sum of timestamps
- Delete old microclusters/merging microclusters if their timestamps are close to each other

Sliding Window Clustering

- Timestamp of the most recent data point added to the vector
- Maintain only the most recent T microclusters

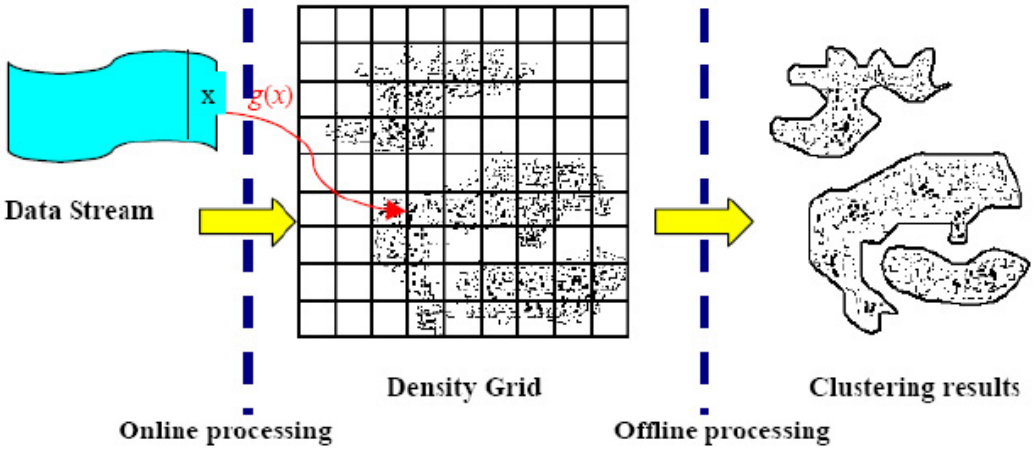
DenStream

- Microclusters are associated with weights based on recency
- Outliers detected by creating separate microcluster

ClusTree

- Allows real-time clustering

Grids



D-Stream

- Assign the data to grids
- Grids are weighted by recency of points added to them
- Each grid associated with a label

DGClust

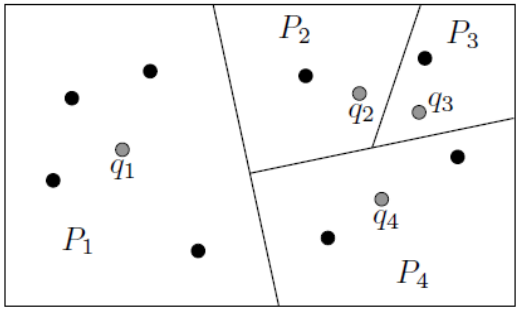
- Distributed clustering of sensor data
- Sensors maintain local copies of the grid and communicate updates to the grid to a central site

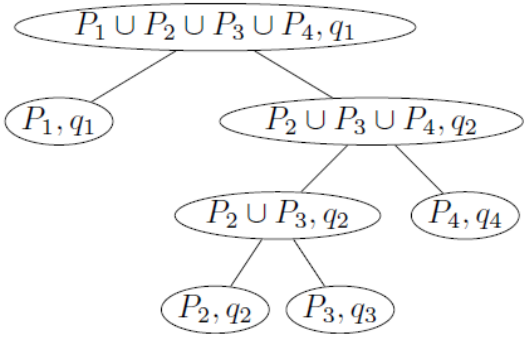
StreamKM++ (Coresets)

Computes a small weighted sample of the data stream and solves the problem on the sample using k-means++ *

- A weighted set S is a (k, ε) coreset for a data set D if the clustering of S approximates the clustering of D with an error margin of ε
  - $(1-\varepsilon) * dist(D,C) \leq dist_w(S,C) \leq (1+\varepsilon) * \text{dist(D, C)}$

- Maintain data in buckets $B_1, B_2 \ldots B_L$.
- Buckets B_2 to B_L contain either exactly 0 or m points.
- B_1 can have any number of points between 0 to m points.
- Merge data in buckets using coreset tree.





merge & reduce

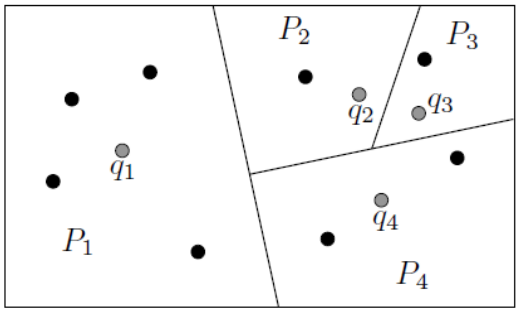
(k-means++: seeding procedure for k-means guaranteing a solution with certain quality, needs random access)

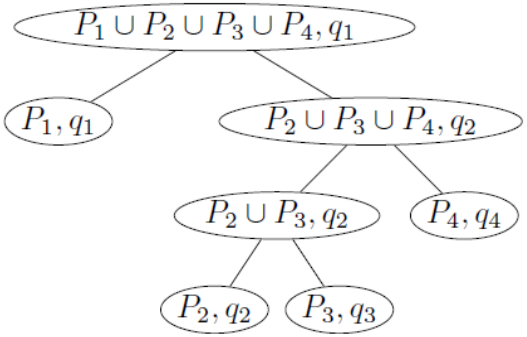
StreamKM++: A Clustering Algorithm for Data Streams, Ackermann, *Journal of Experimental Algorithmics* 2012

StreamKM++ (Coresets)

With each node *u* of coreset tree *T* we store:

- a point set P_u: the cluster associated with node *u*
- a representative point q_u from P_u: obtained by sampling according to d^2 from P_u
- an integer *size(u)*: # of points in set P_u and
- a value *cost(u)*:
  - If *u* is a leaf node, *cost*(P_u, q_u) = sum of squared distances over all points in P_u to q_u
  - If *u* is an inner node, *cost(u)* = sum of cost of its children





merge & reduce

CLASSIFICATION IN DATA STREAMS

Problems In Changing Environment

Introduction

- **Example Problem:** "Finding polarity in feeds from Twitter messages in real time"

- Find text polarity → Text emotion analysis → Text categorization problem
  - Find text polarity → Problem categorizing text into "positive" or "negative"

- Text feeds → System works in real time → Categorize feeds

➢ So, the two main branches of the problem:

  - **Text emotion analysis**
  - **Categorization/classification of data streams**

Data Stream Classification – Challenges

- **Concept drift** → dynamic environment → adaptation to changes , model update
  - Changes in the description of classes – targets
  - Changes in the data distribution
- **Novel Classes** → new classes appear → automated detection
  - Unlabeled data → Outlier detection
- **Limited storage space** → infinite length of data streams, huge amount of data
- **Limited processing time** → high data flow rate → one pass through the data

➢ ***Solutions: incremental learning or learning with multiple classifiers***

Issues: Concept Drift

- Concept Drift: The first and foremost problem in classifiers with changing environments
  - The problem is defined differently as time passes.
  - New features are entering the old space, new features are more important.
  - Distributions change:
    - *Prior probabilities for the c classes, $P(\omega 1),\ldots\ldots P(\omega c)$*
    - *Class- conditional probability distributions, $P(x|\omega i)$, $i= 1,..,c$*
    - *Posterior probabilities $P(\omega i|x)$, $i=1,\ldots,c$*
- Two techniques for dealing with Concept Drift
  - Incremental Learning
  - Ensemble Learning

Incremental Learning (or Online Learning)

- ONLY ONE classifier - enters the learning process every time new data arrives
- Various Incremental Learning Classifiers:
  - Incremental decision tree, Incremental Bayesian algorithm, Incremental SVM, Online Neural Network etc.
- New data arrive instance-by-instance or block-by-block (batches logic is followed)
- Initially the classifier is trained with the first batch of data that arrives

Incremental Learning (or Online Learning)

- As new batches arrive
  - space of features grows (space of classes remains constant)
  - new features enter the space
  - new description of the classes -> concept drift
- Naive probabilities are updated
- As new unlabeled data arrive for testing in batches ***(from classes already seen, NOT new classes)***
  - They are classified
  - We choose the ones that give us the most information (for dealing with Concept Drift)
  - Update the classifier if the system accuracy is at the desired levels

Ensemble Learning

- Multiple learners combining their predictions
- Many online ensemble techniques proposed for changing environments:
  - **Dynamic Combiners:**
    - individual classifiers trained from the beginning
    - changes in the environment -> changes in the *combination rule* (Horse Racing Ensemble Classifiers)

    Cons: Classifiers do not re-train, so they do not adapt to the ever-changing environment

Ensemble Learning (cont'd)

- **Updating the Ensemble Members:** online classifiers updated incrementally in batch mode as new data arrives
  - Same logic with Incremental Learning - there are many classifiers here
    - Ensemble can be derived from different algorithms:
      - » Incremental SVM, Incremental Bayes, Incremental decision tree..
    - Or be separated based on the batches, e.g., if the batches come once a day there may be an ensemble for each day.

- **Dynamic Changes of the line-up of Ensemble:**
  - Individual classifiers dynamically evaluated
  - Worst classifier replaced by a new which has been trained with the latest batch.

Cons: Constantly forgets past data, at some point the old classifiers will all have been replaced and the old knowledge will have been lost.

Ensemble vs Incremental Learning

- For massive data streams we are interested in models that are simplistic
  - Not enough time to run and renew an ensemble
    - depends on the Incremental algorithm to be used, eg an ensemble classifier is faster than an Online Decision Tree

- When time is not so important and accuracy is required then an Ensemble Classifier is the best solution

Unlabeled Data

- Incremental Classifiers & Online Ensembles: incoming Unlabeled Data can be exploited and information extracted
  - New Unlabeled Data is essentially testing data
    - This **data is classified and fed back** to the classifier to address the concept drift problem
- Unlabeled Data are different snapshots of the classes that the Classifier already knows
  - Q: New class? How to detect?
    - Novelty Detection: find outliers

Learn To Forget

- Based on Incremental logic the classifier has the ability to be constantly updated (learn on-line)
  - Any time we stop the classifier we must have the best possible accuracy
    - As new data comes in they define the problem differently and **old data should contribute less**.
- The rate at which the classifier will forget the old data must be chosen correctly
  - match the rate and type of changes made as new data arrives

Learn To Forget

- Use of Windows
  - **Forgetting By Ageing at a Constant Rate:** Forgets at a steady pace, replacing old with new data
  - **Forgetting By Ageing at a Variable Rate:** When a change is detected then the window size changes

- Without the use of windows
  - Integrate age in each instance
    - The more time passes, the less it contributes to the final result
    - The classifier remembers all data with different weights each

Example: Online Classification of Tweets

INCREMENTAL NAÏVE BAYES

Preprocessing of Data

- Tweets: 160 characters long
- Come in batches of 10 sec
  - In each tweet of a batch of data we apply the followins:
    - Remove stop words e.g. *'and'*, *'the'*, *'she'* , etc
    - Remove punctuation marks, hashtags etc
    - Stemming using ***Stanford Core NLP*** , e.g. *"looooovee" and "loved" after* stemming become *"love"*
    - Remove duplicate tokens
    - Maintain emotion-based tokens based on ***SentiWordNet Lexicon***

- Example (batch of tweets)
  - Before preprocessing

    "@Mike, I HATE dental clinics , are so scary places."

    "@Gabriel#cinema My friends and I loved the movie yesterday. It was amazing!!!."

    "I love the nerdy Stanford human biology videos - makes me miss school."

  - After Preprocessing

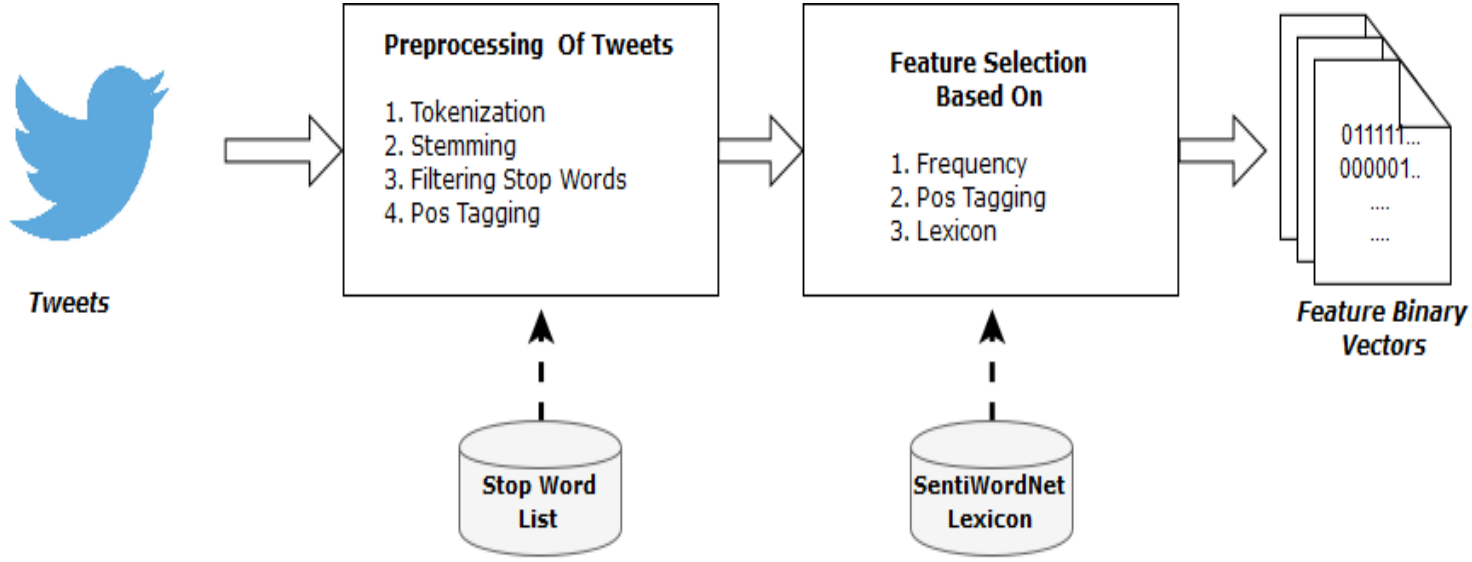
    hate dental clinics scary places

    friend love movie yesterday amazing

    love nerdy human biology video make miss school

Feature Extraction

- After preprocessing features are extracted
- Follow ***bag of words*** *logic*
  - Extract n-grams from each tweet that has been pre-processed
    - 1-gram and 2-grams (to treat cases such as "not love", "not hate"
- The vectors used for the training of the classifier follow ***binary valued*** logic:
  - **1**: if the feature exists in the specific tweet
  - **0**: otherwise.



Feature Extraction

- Suppose the following batch of Tweets arrives as a stream.

  Tweet 1: *"@Mike, I HATE dental clinics, are so scary places."*

  Tweet 2: *"@Gabriel#cinema My friends and I didn't like the movie yesterday!!!!"*

  Tweet 3: *"I love the nerdy Stanford human biology videos -makes me miss school."*

- After Preprocessing:

  Tweet 1: *hate dental clinic scary place*

  Tweet 2: *friend like_NOT movie__NOT*

  Tweet 3: *love nerdy human biology video make miss school*

Features for the batch

F_1	*hate_VERB*
F_2	*dental_ ADJ*
F_3	*clinic_ ADJ*
F_4	*scary_ ADJ*
F_5	*place_NOUN*
F_6	*friend_NOUN*
F_7	*like_NOT_VERB*
F_8	*movie_NOT__NOUN*
F_9	*love_VERB*
F_{10}	*nerdy_ADJ*
F_{11}	*human_ADJ*
F_{12}	*biology_NOUN*
F_{13}	*video_NOUN*
F_{14}	*make_VERB*
F_{15}	*miss_VERB*
F_{16}	*school_NOUN*

Number Of Tweet	Binary-Valued Vector
1	1111100000000000
2	0000011100000000
3	0000000011111111

Recap: Classification of Data Streams - Methods

- ***Incremental Learning*** → we do not have all the data from the beginning
  - Constantly updated as new data arrive (Any time learning)
  - Lossless classifier
  - Learn to forget
  - Decision trees, Naive Bayes, SVMs -> easily adapt to incremental logic

  (Fast, simple)

- ***Ensemble Learning*** → multiple classifiers, combination of predictions
  - **Dynamic combination** → training a priori, changes in the environment are reflected as changes in the combination rules
  - **Updated training set** → classifiers are updated or new are created
    - Reusing stream data *(Online Bagging and Boosting)*
    - *Filtering stream data*
    - *Selecting chunks of data from the stream*
  - **Update of members of the ensemble model**
  - **Structural changes of the ensemble model** → replace older or obsolete
  - **Introduction of new features**

  (Time consuming process, better accuracy)

Incremental Naïve Bayes (labeled data)

- As new labeled data arrive
  - The feature space grows when new batches of tweets come with new features

Example: New tweet in new batch " @elis#travel…I really like traveling!!!!

Feature space before the new batch

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17
hate	dental	clinic	scary	place	friend	love	movie	yesterday	amazing	nerdy	human	biology	video	make	miss	school

New feature space: 2 new features were added. Retain all previous features too.

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
hate	dental	clinic	scary	place	friend	love	movie	yesterday	amazing	nerdy	human	biology	video	make	miss	school	like	travel

  - Update ***prior and conditional probabilities*** every time new *batches* arrive
    - *Multinomial distribution*

Incremental Naïve Bayes (unlabeled data)

- Unlabeled data that arrive are *considered as testing data* and they also come in batch format
  - IMPORTANT: they belong to classes that the classifier has already seen during the training process, they simply appear because they define the problem in a different way (concept drift) and we want to export this information
  - Once the unlabeled data arrives (One of the suggested solutions)
    - Calculate the probability of each element belonging to all possible classes of the classifier
    - If the probability of the two most probable classes satisfies the requirements then the class to which the item belongs is identified by the classifier
    - Otherwise use knn to find the class in conjunction with Naïve Bayes

Concept Drift Management

- Data is streamed -> Naive Bayes requires update of attribute and class statistics

➢ **Choose an incremental Naive Bayes**

- No prior knowledge of the data in the stream
- New Tweets constantly arrive -> new words that describe equally well the two classes arrive -> the description of the target class changes
  - The classifier must detect and adapt to changes in real time

➢ **Choose a dynamic feature space**

- New predictive features are added every time new Tweets arrive
- Feature space high dimensional -> use “learn to forget” idea

Concept Drift Management

- Let us assume that a new batch with one Tweet arrives:

  Tweet 4: *"@elis#travel…I really like traveling!!!!"*

- After Preprocessing:

  Tweet 4: *really like travel*

F_1	*hate_VERB*
F_2	*dental_ ADJ*
F_3	*clinic_ ADJ*
F_4	*scary_ ADJ*
F_5	*place_NOUN*
F_6	*friend_NOUN*
F_7	*like_NOT_VERB*
F_8	*movie_NOT__NOUN*
F_9	*love_VERB*
F_{10}	*nerdy_ADJ*
F_{11}	*human_ADJ*
F_{12}	*biology_NOUN*
F_{13}	*video_NOUN*
F_{14}	*make_VERB*
F_{15}	*miss_VERB*
F_{16}	*school_NOUN*
F_{17}	*really_ADV*
F_{18}	*like_VERB*
F_{19}	*travel_NOUN*

Number Of Tweet	Binary-Valued Vector
4	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1

Naïve Bayes: Three versions

- *Incremental Nave Bayes with dynamic feature space*
- *Self-trained incremental Naive Bayes with dynamic feature space*
- *Co-trained incremental Naive Bayes with dynamic feature space*

Incremental Naive Bayes

- Based on static Naive Bayes → Assuming multinomial distribution for finding text polarity we have:
  - The probability of a tweet d belonging to a class c is given by:

$$P(c \mid d) \propto P(c) * \prod_{1 \le k \le n_d} P(t_k \mid c)$$

    - ***P* (t_k | *c*):** cond. prob of a token t_k to appear in text d of class c.
    - ***P* (*c*)**: a priori prob. of text to be in class c.
    - **< t_1, t_2, . . . , t_k >** : tokens of Tweet d, that are parts of the vocabulary used in classification.
    - **n_d** : # of tokens in a Tweet d.
  - Goal is finding the best class for the tweet, i.e, the most probable class
    - Maximum a posteriori (MAP) class C_{map}

$$C_{\text{map}} = \text{argmax}_{c \in C} \hat{P}(c \mid d) = \text{argmax}_{c \in C} \hat{P}(c) * \prod_{1 \le k \le n_d} \hat{P}(t_k \mid c)$$

Incremental Naive Bayes

– Once the first batch with the labeled Tweets arrives and the features are extracted, the classifier is trained by calculating the ***prior*** *and* ***conditional probabilities***:

$$\hat{P}(c) = \frac{N_c}{N}$$

where N_c is the number of tweets in class C and N number of total tweets

$$\hat{P}(t \mid c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

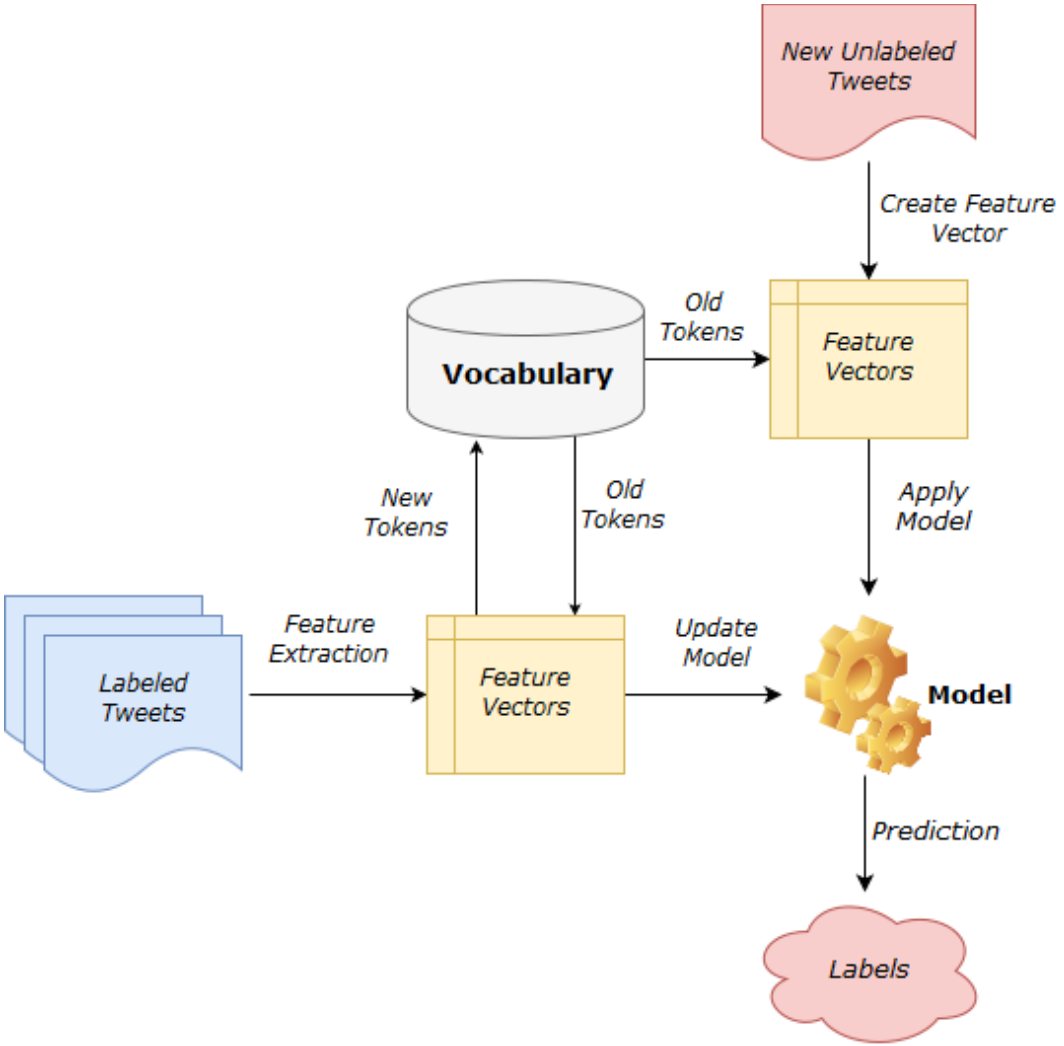
where T_{ct} is the number of occurrences of token t of class C in training tweets
Ratio: sum of occurrences of each token t of the set of features in tweets of class c

Incremental Naive Bayes

- Naive Bayes easily converts to incremental:
  - Renew the following for new data:
    - **N** → Total # of Tweets that have reached the system since the beginning of its operation.
    - $\mathbf{N}_c$ → # of texts Τον αριθμό των κειμένων belonging to class c, for each class.
    - $\mathbf{T}_{ct}$ → # of occurrences of token t in texts of class c, for each class and each token.
    - **Vocabulary** → feature space is updated with new tokens as new Tweets arrive.

➢ Therefore, incremental Naive Bayes with dynamic feature space:
  - Small space, as long as it remembers the above values in memory
  - Updates old values of prior and conditional probabilities with a **single pass of data**
  - Faces concept-drift with dynamic feature space and by updating statistics in each iteration
  - Simple: just increase the above values for new Tweets
  - Does not need to know the correlations of the features

Incremental Naive Bayes-Architecture



Unlabeled Tweets

- So far we assume that data reaching the stream is labelled..
  - Not true!
- Tweets arrive in the form of a feed from the Twitter API
  - They are written by individuals on a topic -> they do not contain labels
- Manual categorization of Tweets
  - Time consuming process -> there is no time to process data
  - High cost

➢ **Need to find a solution to deal with unlabeled Tweets !!!**

Self-trained incremental Naive Bayes

- ***Combination of incremental logic and semi-supervised learning***

- **Self-training logic** → The classifier uses its own predictions to label Tweets
  - Once the data has labels it can be inserted into the training set of the incremental classifier and update it
- Problem change: Can the classifier "learn" from the data it classified itself?
  - i.e., how well it can distinguish the class "positive" from "negative"
  - Selection of the most reliable predictions based on the model -> in Naive Bayes, the class with the highest probability is selected

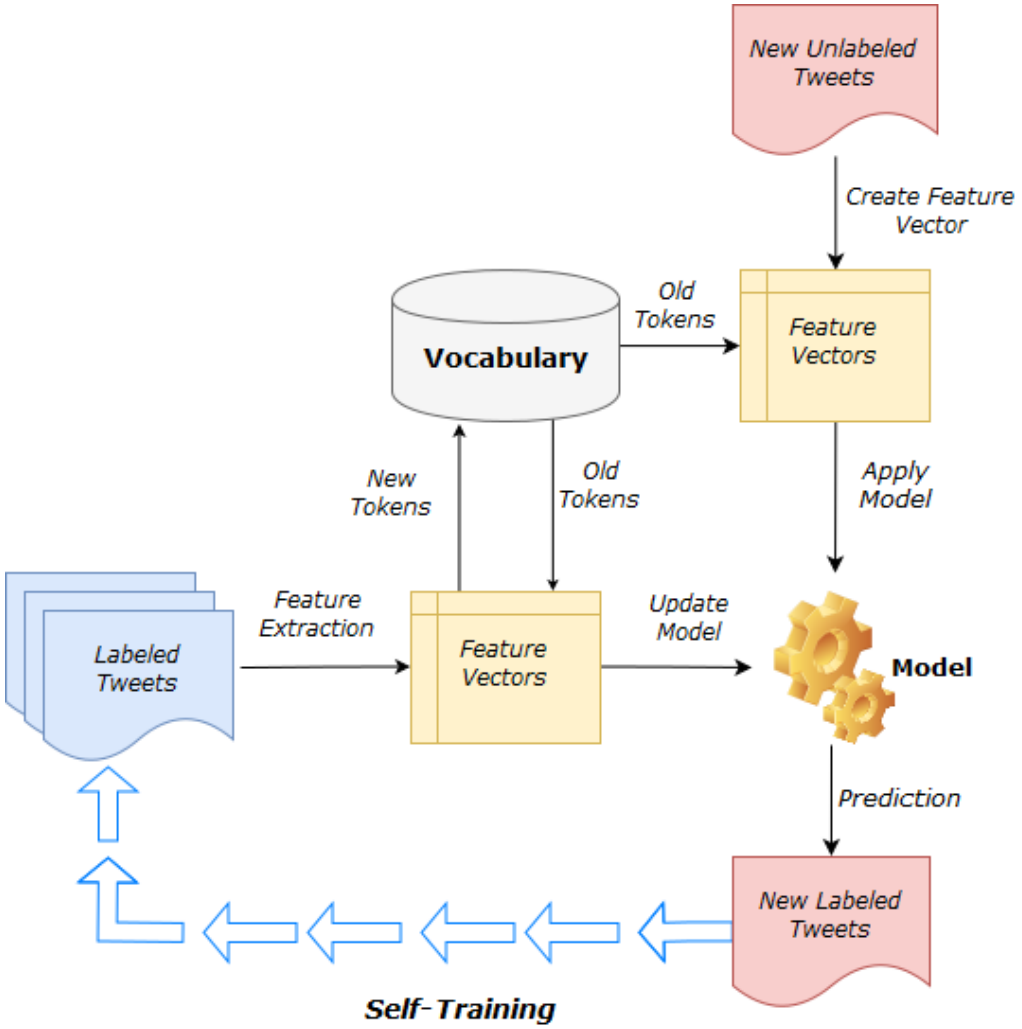
➢ Most reliable predictions:

$$P(\text{positive} \mid \text{tweet}) \geq 2 * P(\text{negative} \mid \text{tweet})$$

or

$$P(\text{negative} \mid \text{tweet}) \geq 2 * P(\text{positive} \mid \text{tweet})$$

Self-trained incremental Naive Bayes-Architecture



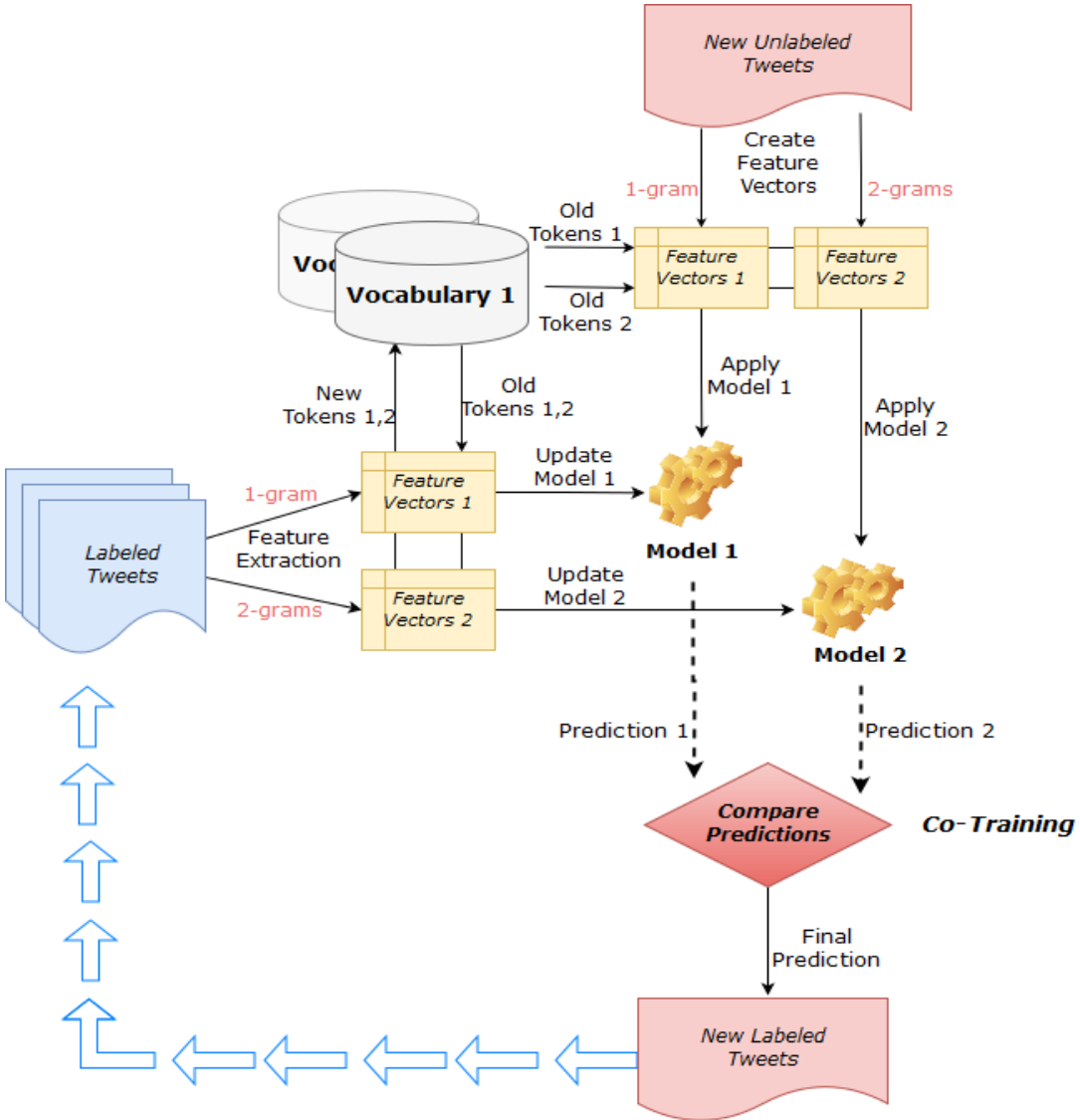
Co-trained incremental Naive Bayes

- ***Combination of incremental logic and semi-supervised learning***

- **Co-training logic** → feature space is divided into two independent sets and two independent classifiers operate simultaneously
  - Incremental Naive Bayes and Self-trained inremental Naive Bayes → 1-gram → fast extraction
  - Co-training → extraction of 2-grams
    - Pairs of words are identified e.g. adjective-noun, adverb-adjective etc.
    - Independent set from the 1-grams
- Two incremental Naive Bayes with dynamic feature space are selected to operate simultaneously
  - First model is based on 1-gram, second on 2-grams
- For unlabeled Tweets
  - Each model makes its own predictions based on characteristics extracted
  - Predictions of the two classifiers compared, when agree, introduce into the training set
  - Both incremental Naive Bayes renewed.

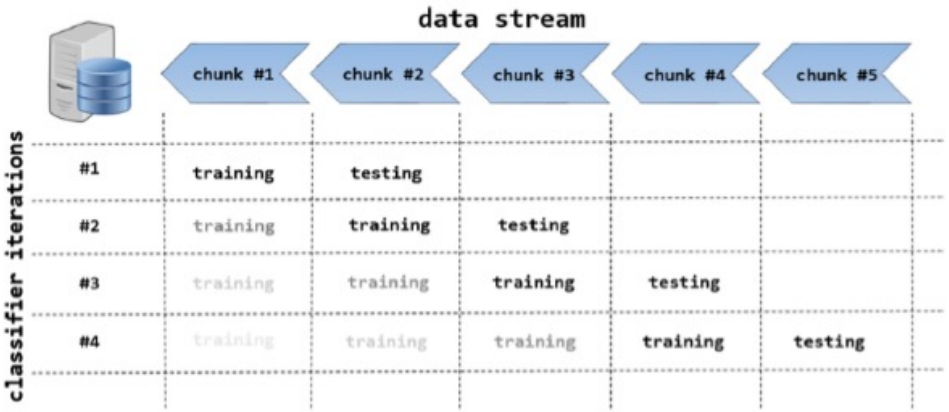
➢ **Essentially, one classifier learns the other and the decisions are joint**

Co-trained incremental Naive Bayes-Architecture



Experiments

- *Simulation of Twitter API*
  - Data from **Sentiment140** . CSV file with > 1.000.000 Tweets with the following fields:
    - **0 – Tweet polarity (0 = negative, 4 = positive)**
    - 1 – Tweet ID (e.g. 2087)
    - 2 – Tweet date and time (e.g. Sat May 16 23:58:44 UTC 2009)
    - 3 – the query (π.χ. lyx). If there is no query, then value is NO_QUERY.
    - 4 – user (e.g. robotickilldozr)
    - **5 – Tweet text (e.g.. Lyx is cool).**
- All classifiers are trained with a total of 60,000 Tweets that were sampled from the data set -> flow is activated -> remaining Tweets enter the flow every 30sec = batch duration
- *Evaluation*
  - **Phase 1**: After each batch of Tweets reach the stream we evaluate based on the Tweets of the test set that exists in Sentiment140
  - **Phase 2**: After each batch we evaluate with next batch that arrives -> *results depend on time t and the order that the batches arrive*
  - ***Thus we randomly select a time point t and we perform the experiment for up to t + 10 batches***



Experiments: Evaluation

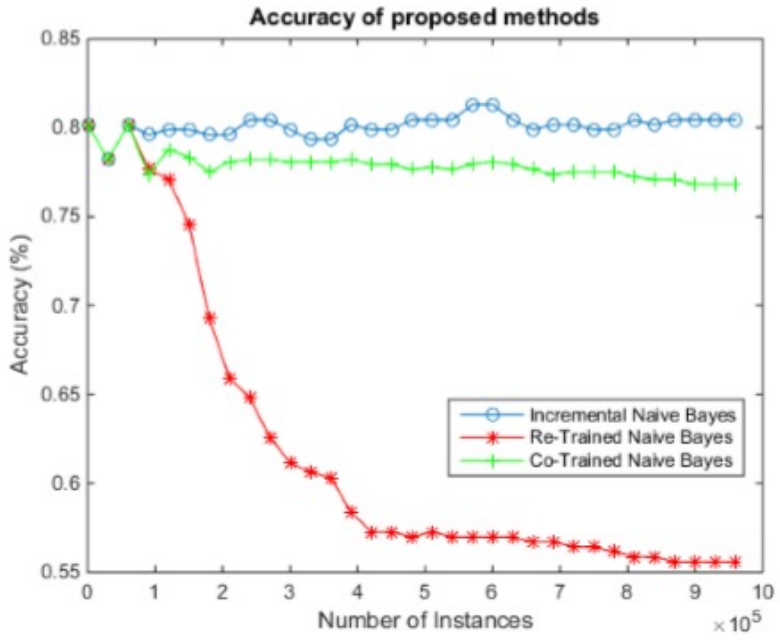
$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN},$$

- TP (True Positives), FP (False Positives), TN (True Negatives), FN (False Negatives)

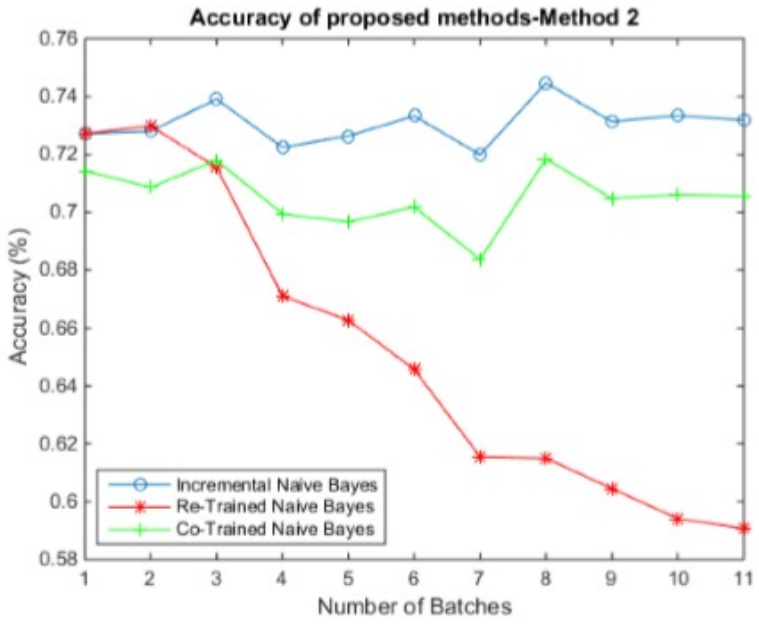
- Time duration in sec from the moment a new batch reaches the flow until the model is renewed by this new batch

- All three classifiers are trained offline with the dateset of 60.000 Tweets.
  Then:
  - The incremental **Naïve Bayes** accepts batches of **labeled Tweets**
  - The **Self-trained** and **Co-trained** accept batches of **unlabeled Tweets**

Results

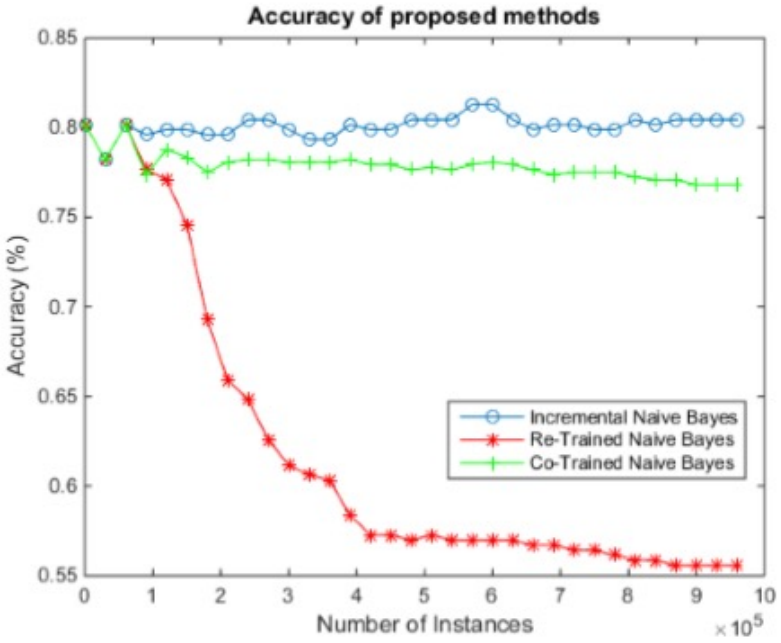


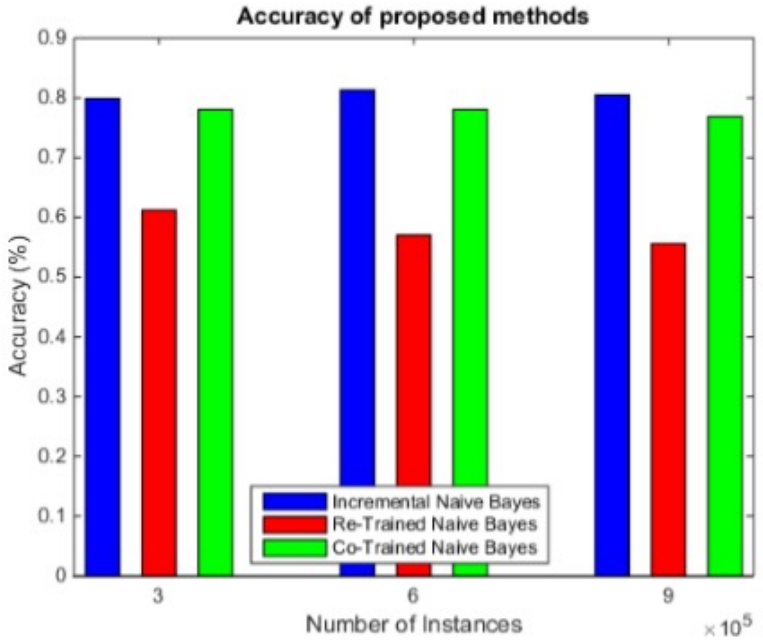
Phase 1



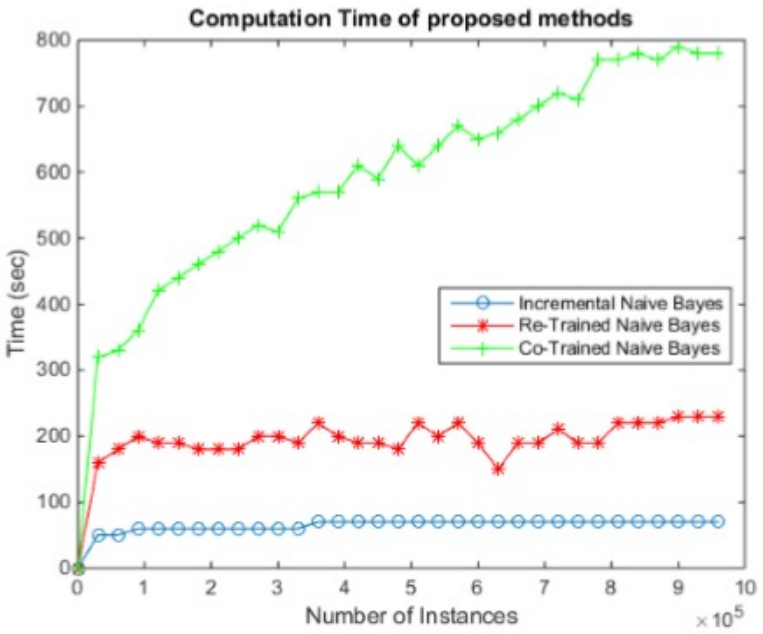
Phase 2

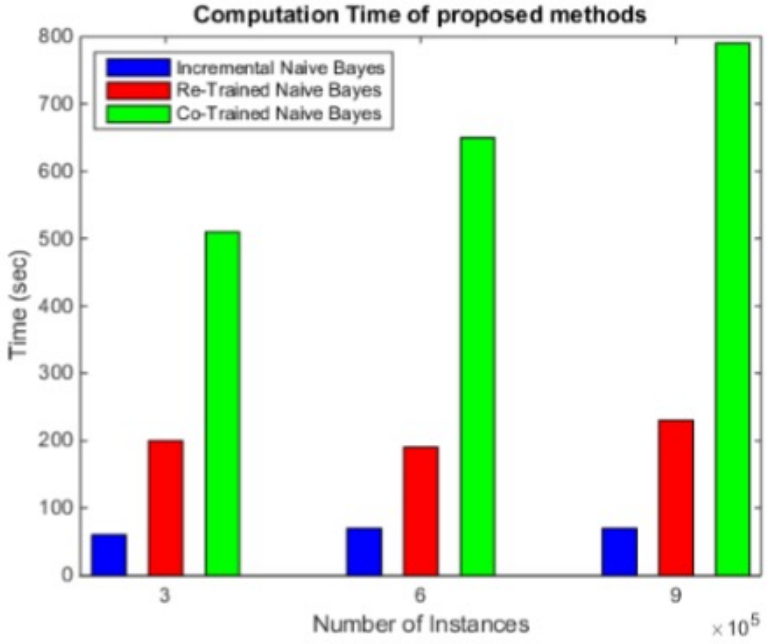
Results


Accuracy of proposed methods
Accuracy (%)
Number of Instances
×10^5
Incremental Naive Bayes
Re-Trained Naive Bayes
Co-Trained Naive Bayes


Accuracy of proposed methods
Accuracy (%)
Number of Instances
×10^5
Incremental Naive Bayes
Re-Trained Naive Bayes
Co-Trained Naive Bayes

Results


Computation Time of proposed methods
Time (sec)
Number of Instances
$\times 10^5$
Incremental Naive Bayes
Re-Trained Naive Bayes
Co-Trained Naive Bayes


Computation Time of proposed methods
Time (sec)
Number of Instances
$\times 10^5$
Incremental Naive Bayes
Re-Trained Naive Bayes
Co-Trained Naive Bayes

Conclusions

- Best performance Incremental Naïve Bayes, then Co-trained Incremental Naïve Bayes, then Self-trained Naïve Bayes
- Incremental Naïve Bayes and Co-trained incremental Naïve Bayes
  - consistent behavior
  - good class separation
  - good selection of confident items for Co-trained incremental Naïve Bayes over Self-trained
- Incremental Naïve Bayes and Co-trained incremental Naïve Bayes remember what they have learned as time passes and new knowledge arrives (1st phase of experiment)
  - Not the same for Self-trained Naïve Bayes

Conclusions

- Incremental Naïve Bayes and Co-trained incremental Naïve Bayes try to adapt to the changes, adapt to the concept drift not so good as the accuracy does not increase (2nd phase of experiment)
  - The choice and the large space of the features are responsible and not the system
  - Self-trained Naïve Bayes can not learn new information
    - Nature of the system responsible, an error in prediction has major effect in subsequent iterations
- Incremental Naïve Bayes and Co-trained incremental Naïve Bayes similar performance
  - Co-trained incremental Naïve Bayes has similar performance with a classifier trained on labeled data
  - Incremental Naïve Bayes has issues on feature selection and number of features

Αλγόριθμος	**Μ.Ο Ακρίβειας(φ1)**	**Μ.Ο Ακρίβειας(φ2)**	**Μ.Ο Χρόνου**
Αυξητικός Naïve Bayes	80%	73%	70sec
Self-Trained Αυξητικός Naïve Bayes	↘ πτωτική συμπεριφορά	↘ πτωτική συμπεριφορά	200sec
Co-Trained Αυξητικός Naïve Bayes	78%	71%	↗ αυξανόμενη συμπεριφορά

References

[1]. Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas, On the Utility of Incremental Feature Selection for the Classification of Textual Data Streams, 2005, Spinger

[2]. CHEN hua, ZHANG xiao-gang, ZHANG Jing, Ding Li-hua, A Simplified Learning Algorithm of Incremental Bayesian, 2009, World Congress on Computer Science and Information Engineering

[3]. Sotiris Kotsiantis, Increasing the Accuracy of Incremental Naive Bayes Classifier Using Instance Based Learning, 2013, International Journal of Control, Automation, and Systems

[4]. Frank Klawonn, Plamen Angelov, Evolving Extended Naive Bayes Classifiers

[5]. Ludmila I. Kuncheva, Classifier Ensembles for Changing Environments, Proc. 5th Int. Workshop on Multiple Classifier Systems, Cagliari, Italy, Springer-Verlag, LNCS, 3077, 2004, 1–15

[6]. Parneeta Sidhu M. P. S. Bhatia, An online ensembles approach for handling concept drift in data streams: diversified online ensembles detection, 2015, Springer-Verlag Berlin Heidelberg

[7]. Shuxia Ren, Yangyang Lian, Xiaojian Zou, Incremental Naïve Bayesian Learning Algorithm based on Classification Contribution Degree, JOURNAL OF COMPUTERS, VOL. 9, NO. 8, AUGUST 2014

[8]. Haixun Wang, Wei Fan, Philip S. Yu, Jiawei Han, Mining Concept-Drifting Data Streams using Ensemble Classifiers

[9]. Bartosz Krawczyk, Michał Wozniak, One-class classifiers with incremental learning and forgetting for data streams with concept drift, 2014, Springer

[10]. Bartosz Krawczyk, Michał Wozniak, Weighted Naive Bayes Classifier with Forgetting for Drifting Data Streams, 2015 IEEE International Conference on Systems, Man, and Cybernetics

[11]. Wenyu Zang, Peng Zhang, Chuan Zhou, Li Guo, Comparative study between incremental and ensemble learning on data streams: Case study, Zang et al. Journal of Big Data 2014

[12]. Moharned Medhat Gaber, Arkady Zaslavsky, Shonali Krishnaswamy, A SURVEY OF CLASSIFICATION METHODS IN DATA STREAMS

References

[1]. Big Data Sentiment Analysis for Brand Monitoring in Social Media Streams by Cloud Computing, Francesco Benedetto and Antonio

Tedeschi, W. Pedrycz and S.-M. Chen (eds.), Sentiment Analysis and Ontology Engineering,

Studies in Computational Intelligence 639, Springer International Publishing Switzerland, pages 341-377, 2016

[2]. Speech and Language Processing. Daniel Jurafsky & James H. Martin, Chapter 7: Classification: Naive Bayes,Logistic Regression, Sentiment, Alan Apt, Prentice Hall, Englewood Cliffs, New Jersey 07632, 2015

[3]. A Comparative Study on Sentiment Analysis, Alireza Yousefpour, Roliana Ibrahim, Haza Nuzly Abdull Hamed, Mohammad Sadegh Hajmohammadi, Advances in Environmental Biology, 8(13), AENSI Journals, pages 53-68, 2014

[4]. Active Learning Literature Survey,Burr Settles,Computer Sciences Technical Report 1648 University of Wisconsin–Madison Updated on: January 26, 2010

[5]. Adaptive Semi Supervised Opinion Classifier With Forgetting Mechanism, Max Zimmermann, Eirini Ntoutsi, Myra Spiliopoulou, Proceeding SAC '14 Proceedings of the 29th Annual ACM Symposium on Applied Computing, SAC'14 March 24-28, 2014, Gyeongju, Korea, ACM, pages 805-812, 2014

[6]. An adaptive ensemble classifier for mining concept drifting data streams, Dewan Md. Farid, Li Zhang, Alamgir Hossai, Chowdhury Mofizur Rahman,Rebecca Strachan, Graham Sexton, Keshav Dahal, Expert Systems with Applications 40, Elsevier, pages 5895–5906, 2013

[7]. A Practical Approach to Classify Evolving Data Streams: Training with Limited Amount of Labeled Data, Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, Data Mining, 2008. ICDM '08. Eighth IEEE International Conference, IEEE, 2008

[8]. A Simplified Learning Algorithm of Incremental Bayesian, CHEN hua, ZHANG xiao-gang, ZHANG Jing, Ding Li-hua, Computer Science and Information Engineering, 2009 WRI World Congress, IEEE, 2009

[9]. A Streaming Ensemble Algorithm (SEA) for LargeScale Classification, W. Nick Street, YongSeog Kim, KDD '01 Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining,San Francisco, California, ACM New York, pages 377-382, 2001

[10]. A survey of open source tools for machine learning with big data in the Hadoop ecosystem, Sara Landset, Taghi M. Khoshgoftaar, Aaron N. Richter, Tawfiq Hasanin, A.N. et al. Journal of Big Data 2: 24, doi:10.1186/s40537-015-0032-1, 2015

[11]. Challenges in Sentiment Analysis, Saif M. Mohammad, A Practical Guide to Sentiment Analysis 2015, National Research Council Canada, 2015

[12]. Classifier Ensembles for Changing Environments, Ludmila I. Kuncheva, Published in: F. Roli, J. Kittler and T. Windeatt (Eds.), Proc. 5th Int. Workshop on Multiple Classifier Systems, Cagliari, Italy, Springer-Verlag, LNCS, 3077, pages 1–15, 2004

[13]. Combining Lexicon and Machine Learning Method to Enhance the Accuracy of Sentiment Analysis on Big Data 1G. Vaitheeswaran, 2Dr.Arockiam, G. Vaitheeswaran et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (1), pages 306-311, 2016

References

[14]. Comparative study between incremental and ensemble learning on data streams: Case study, Wenyu Zang, Peng Zhang, Chuan Zhou, Li Guo, , Zang et al. Zang et al. Journal of Big Data 2014, 1:5, Springer, 2014

[15]. Competitive Self-Training Technique for Sentiment Analysis in Mass Social Media, Sola Hong, Jaedong Lee, Jee-Hyong Lee, SCIS&ISIS 2014, Kitakyushu, Japan, IEEE, 2014

[16]. Co-training for Semi-Supervised Sentiment Classification Based on Dual-view Bags-of-words Representation, Rui Xia, ChengWang, Xinyu Dai, Tao Li, Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, Association for Computational Linguistics, pages 1054–1063, 2015

[17]. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts, Janyce Wiebe, Ellen Riloff, Computational Linguistics and Intelligent Text Processing, Volume 3406 of the series Lecture Notes in Computer Science, Springer, pages 486-497, 2005

[18]. Dynamic classifier ensemble for positive unlabeled text

stream classification, Shirui Pan, Yang Zhang, Xue Li, Knowledge and Information Systems, Volume 33, Issue 2, Springer-Verlag London Limited, pages 267–287, 2012

[19]. Dynamic Feature Space and Incremental Feature Selection for the Classification of Textual Data Streams, Ioannis Katakis , Grigorios Tsoumakas , Ioannis Vlahavas, ECML/PKDD-2006 International Workshop on Knowledge Discovery from Data Streams. 2006

[20]. Evolving Extended Naive Bayes Classifiers, Frank Klawonn, Plamen Angelov, Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference, IEEE, 2006, [21]. Exploring Feature Definition and Selection for Sentiment Classifiers,Yelena Mejova, Padmini Srinivasan, Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, Association for the Advancement of Artificial

Intelligence, 2011

[21]. Facing the reality of data stream classification: coping with scarcity of labeled data, Mohammad M. Masud, Clay Woolam, Jing Gao, Latifur Khan, Jiawei Han, Kevin W. Hamlen, Nikunj C. Oza, Knowledge and Information Systems, Volume 33, Issue 1, Springer, pages 213–244, 2012

[22]. Feature Selection for Twitter Sentiment Analysis: An Experimental Study, Riham Mansour, Mohamed Farouk Abdel Hady, EmanHosam, Hani Amr, Ahmed Ashour, Computational Linguistics and Intelligent Text Processing,Volume 9042 of the series Lecture Notes in Computer Science, Springer, pages 92-103, 2015

[23]. A Survey on Concept Drift Adaptation, JOAO GAMA, INDRE ZLIOBAITE, ALBERT BIFET, MYKOLA PECHENIZKIY, ABDELHAMID BOUCHACHIA, Bournemouth University, UK ACM Computing Surveys, Vol. 1, No. 1, Article 1, 2013

[24]. Fast Data Processing with Spark Second Edition,Krishna Sankar, Holden Karau, Packt Publishing, 2015

References

[25]. Improving Twitter Sentiment Analysis with Topic-Based Mixture

Modeling and Semi-Supervised Training, Bing Xiang, Liang Zhou, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 434–439, Baltimore, Maryland, USA, 2014

[26]. Increasing the Accuracy of Incremental Naive Bayes Classifier Using Instance Based Learning, Sotiris Kotsiantis, International Journal of Control, Automation and Systems, Volume 11, Issue 1, pages 159–166, Sringer, 2003

[27]. Machine Learning and Lexicon based Methods for Sentiment Classification: A Survey, Hailong Zhang, Wenyan Gan, Bo Jiang Published in: ProceedingWISA '14 Proceedings of the 2014 11th Web Information System and Application Conference, September 12 - 14, pages 262-265, IEEE, 2014

[28]. Mining Concept Drifting Data Streams using Ensemble Classifiers, Haixun Wang, Wei Fan, Philip S. Yu, Jiawei Han, ProceedingKDD '03 Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 226-235, ACM, 2003

[29]. Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches, Pimwadee Chaovalit, Lina Zhou, Proceeding HICSS '05 Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 4 - Volume 04, January 03 - 06, pages 112.3, IEEE, 2005,

[30]. NRC-Canada-2014: Recent Improvements in the Sentiment Analysis of Tweets, Xiaodan Zhu, Svetlana Kiritchenko, Saif M. Mohammad, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, August 23-24, pages 443–447, 2014

[31]. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets, Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu, In Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013), Atlanta, USA, arXiv:1308.6242 [cs.CL], 2013

[32]. On Demand Classification of Data Streams, Charu C. Aggarwal, Jiawei Han, Philip S. Yu, ProceedingKDD '04 Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 503-508, ACM, Seattle, WA, USA August 22 - 25, 2004

[33]. One-class classifiers with incremental learning and forgetting

for data streams with concept drift, Bartosz Krawczyk, Michał Wozniak, JournalSoft Computing - A Fusion of Foundations, Methodologies and Applications archive, Volume 19, Issue 12, pages 3387-3400, Springer-Verlag Berlin, Heidelberg, 2015

[34]. Online Learning: Searching for the Best Forgetting Strategy under Concept Drift, Ghazal Jaber, Antoine Cornuéjols, Philippe Tarroux, Neural Information Processing,Volume 8227 of the series Lecture Notes in Computer Science pages 400-408, Springer, 2013

[35]. On the Optimality of the Simple Bayesian Classifier under Zero-One Loss, Pedro Dominigos, Michael Pazzani, JournalMachine Learning - Special issue on learning with probabilistic representations archive, Volume 29, Issue 2-3, pages 103 – 130, Kluwer Academic Publishers Hingham, MA, USA, 1997

[36]. On the Utility of Incremental Feature Selection for the Classification of Textual Data Streams, Ioannis Katakis, Grigorios Tsoumakas, Ioannis Vlahavas, Advances in Informatics, Volume 3746 of the series Lecture Notes in Computer Science, pages 338-348, Springer, 2005

[37]. Opinion Mining and Sentiment Analysis, Bo Pang, Lillian Lee, Journal Foundations and Trends in Information Retrieval archive, Volume 2 Issue 1-2, pages 1-135, ACM, 2008

[38]. Online ensemble learning, Nikunj Chandrakant Oza, Chairs: Stuart Russell, Doctoral Dissertation Online ensemble learning, University of California, Berkeley, ISBN:0-493-58497-8, 2001

[39]. Predicting the Effectiveness of Self-Training: Application to Sentiment Classification, Vincent Van Asch, Walter Daelemans, Computation and Language, arXiv:1601.03288 [cs.CL], 2016

[40]. Self-training from labeled features for sentiment analysis, Yulan He, Deyu Zhou JournalInformation Processing and Management: an International Journal archive, Volume 47 Issue 4, pages 606-616, Pergamon Press, Inc. Tarrytown, NY, USA, Elsevier, 2011

References

[41]. Semi-Supervised Learning Literature Survey, Xiaojin Zhu, University of Wisconsin Madison, July 19, 2008

[42]. Sentiment Analysis: Incremental learning to build domain models

Raimon Bosch, Chairs: Leo Wanner, Master thesis, Universitat Pompeu Fabra, 2013

[43]. Sentiment Analysis: Adjectives and Adverbs are better than Adjectives Alone, Benamara F, Cesarano C, Picariello A, Reforgiato D, Proceedings of the International Conference on Weblogs and Social Media (ICWSM), 2007

[44]. Sentiment analysis algorithms and applications: A survey, Walaa Medhata, Ahmed Hassanb, Hoda Korashyb, Ain Shams Engineering Journal, Volume 5, Issue 4, pages 1093–1113, Elsevier, 2014

[45]. Sentiment Analysis and Opinion Mining, Bing Liu, University of Illinois at Chicago, Morgan & Claypool Publishers, May 2012.

[46]. Sentiment Analysis: Capturing Favorability Using Natural Language Processing, Tetsuya Nasukawa, Jeonghee Yi, Proceeding K-CAP '03 Proceedings of the 2nd international conference on Knowledge capture, pages 70-77, ACM, Sanibel Island, FL, USA, October 23 - 25, 2003,

[47]. Sentiment Analysis in Social Media Texts, Alexandra Balahur, Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 120–128, Atlanta, Georgia, 14 June 2013

[48]. Sentiment Analysis in Twitter using Machine Learning Techniques, Neethu M S, Rajasree R, Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference, Tiruchengode, India,IEEE, 2013

[49]. Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification, Prem Melville, Wojciech Gryc, Richard D. Lawrence, ProceedingKDD '09 Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1275-1284, ACM, Paris, France -June 28 - July 01, 2009,

[50]. Sentiment Analysis of Twitter Data, Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau, ProceedingLSM '11 Proceedings of the Workshop on Languages in Social Media, pages 30-38, ACM,Portland, Oregon, June 23 - 23, 2011

[51]. Sentiment Analysis on Twitter Streaming, DataSanthi Chinthala , Ramesh Mande, Suneetha Manne, Sindhura Vemuri, Emerging ICT for Bridging the Future - Proceedings of the 49th Annual Convention of the Computer Society of India (CSI), Volume 1, Volume 337 of the series Advances in Intelligent Systems and Computing, pages 161-168, 2015

[52]. SESS: A Self-Supervised and Syntax-Based Methodfor Sentiment Classification,*, Weishi Zhang, Kai Zhao, Likun Qiu, Changjian Hu, 23rd Pacific Asia Conference on Language, Information and Computation, City University of Hong Kong, pages 596–605, 2009

[52]. The Stanford CoreNLP Natural Language Processing Toolkit, Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, David McClosky, In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014

[53]. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews, Peter D. Turney, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics(ACL), Philadelphia, July 2002, pages 417-424, ACM, 2002

[54]. Thumbs up? Sentiment Classification using Machine Learning Techniques, Bo Pang, Lillian Lee, Shivakumar Vaithyanathan, ProceedingEMNLP '02 Proceedings of the ACL-02 conference on Empirical methods in natural language processing, Volume 10, pages 79-86, Association for Computational Linguistics Stroudsburg, PA, USA, 2002

References

[56]. Twitter Sentiment Analysis: The Good the Bad and the OMG!, Efthymios Kouloumpis, TheresaWilson, Johanna Moore, Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011. AAAI Press, pages 538-541, 2011

[57]. Twitter Sentiment Classification using Distant Supervision, A. Go, R. Bhayani, L. Huang. Processing 2009

[58]. Using Unlabeled Data to Improve Text Classification, Kamal Paul Nigam, Chairs: Tom M. Mitchell, Phd Thesis, School of Computer Science Carnegie Mellon University, Pittsburgh, 2001

[59]. Online Bagging and Boosting, Nikunj C. Oza , Stuart Russell , In Artificial Intelligence and Statistics, 2001

[60]. Advanced Analytics with Spark, Sandy Ryza, Uri Laserson, Sean Owen, Josh Wills, Published by O'Reilly Media,2015

[61]. Data Classification Algorithms and Applications, Charu C. Aggarwal,CRC Press, 2015

[62]. Getting Started with Apache Spark,From Inception to Production, James A. Scott, Published by MapR Technologies,2015

[63]. Learning Spark,Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia, Published by O'Reilly Media, 2015

[64]. Mastering Apache Spark,Mike Frampton,Published by Packt Publishing Ltd, 2015

[65]. Programming in Scala, Martin Odersky, Lex Spoon, Bill Venners, Artima Press, 2008

[66]. Programming Scala, Second Edition, Dean Wampler, Alex Payne, Published by O'Reilly Media,2015

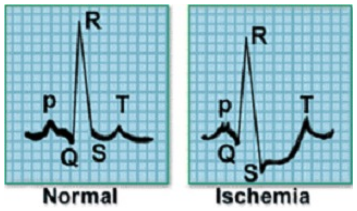
[67]. Scala Cookbook, Alvin Alexander,Published by O'Reilly Media,2013

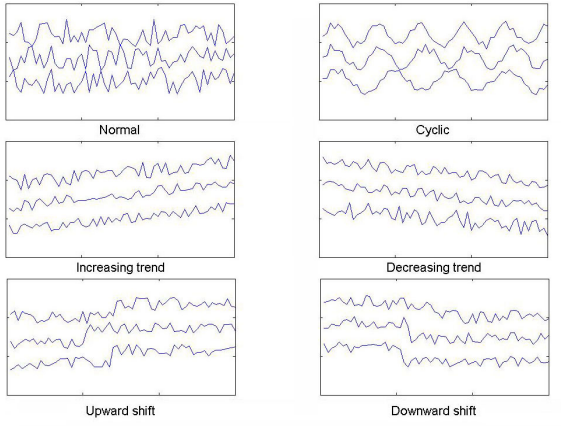
[68]. Machine Learning with Spark, Nick Pentreath, Packt Publishing, 2015

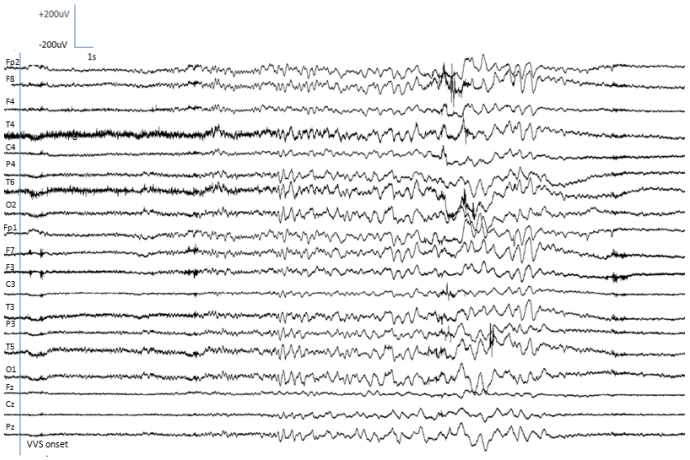
Classification of data streams: EEG, stock trend classification

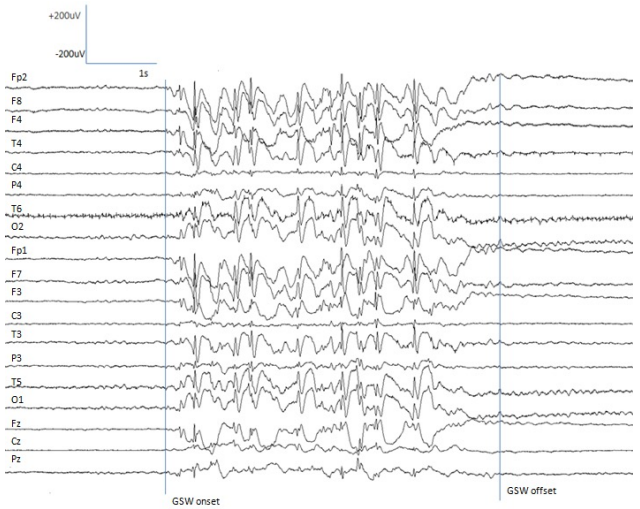
G. Dimitropoulos, E. Papagianni and V. Megalooikonomou, 2017

Classification







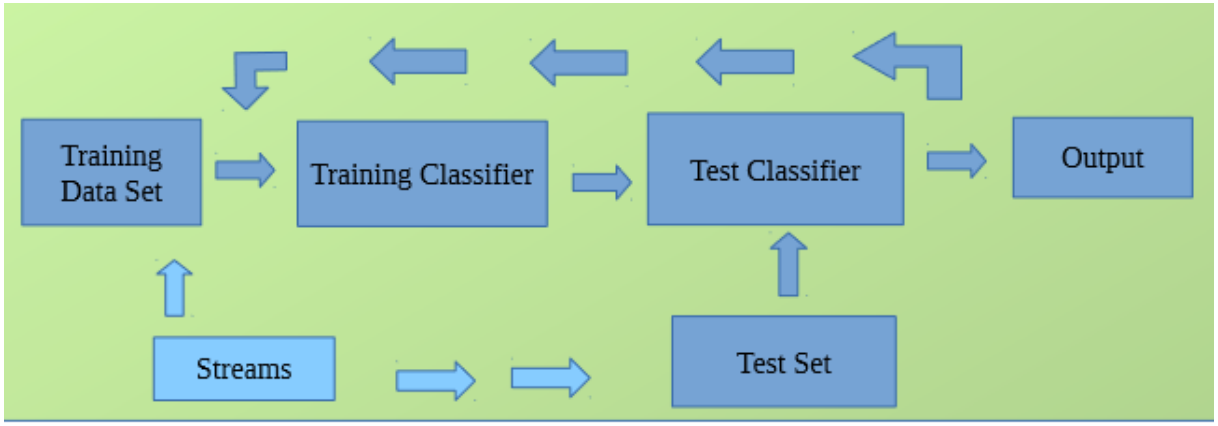


Classification

- Exact incremental learning and adaption of SVM classifiers [Poggio et al. 2001], [Diehl et al. 2003], [Syed et al. 1999].
  - able to learn and unlearn manifold examples
  - adapt the current SVM to changes in regularization and kernel parameters and evaluate generalization performance
- Several applications (e.g., medical and macroeconomic environments)
  - Classification/recognition of certain events in EEG or ECG [Mporas et al. 2015]
  - Classification/recognition of stock trends [Edwards et al. 2007]
  - SVMs for the prediction of trend of the daily Korea Composite Stock Price Index (KOSPI) [Kim 2003]
- This work: improves classification accuracy accomplished by Kim (2003) on the KOSPI dataset by employing the incremental SVM learning algorithm proposed by Diehl et al. (2003)

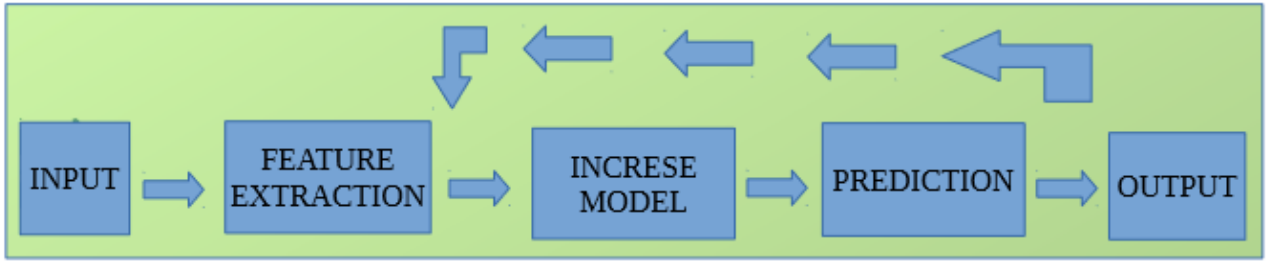
Data Stream Classification

- Incremental SVM learning algorithm [Diehl et al. 2003]
- Model adapts to upcoming data
- Model maintains its existing knowledge through the adaptation of the hyperplane of separation.
- The test set becomes part of the current training set -> increase of model's knowledge.



Data Stream Classification - Stock trends

- Model for stock trend prediction involves as second step feature extraction (12 technical indicators used in [Shin, 2005], [Kim, 2003] etc).
- The classification model is used to predict the trend (upward or downward) the index will have in the next days
- Microsoft StreamInsight (2016) was used to create and handle the streams; Matlab was used to implement the rest of the functions
- Incremental SVM learning algorithm proposed by Diehl et al. 2003 used through the library Diehl (2011) implemented in Matlab



Stock trend classification results (streams)

ALGORITHM	ACCURACY
Classic SVM	52%
Static Incremental SVM	60%
Online SVM	62%
Incremental SVM	**74%**

Incremental SVM: trained on the first 3000 examples then increased its knowledge with feedback from the next 3000 examples (window of size 10 examples). Achieves 71% accuracy on the first half, 77% on the second

Classic SVM: trained with the same first set of 3000 examples and then tested with the next 3000 examples

Static incremental SVM: uses the same training and test sets of 3000 samples each. It uses the SVM incremental learning algorithm instead of the classic SVM. It is only trained once.

Online SVM: trained with the same first 3000 examples; then increases the initial training set with the following 3000 samples; Re-trained every time a new instance arrives

Conclusions

- Incremental SVM learning algorithm for high accuracy classification of streams of data
- Future work:
  - look at classification and clustering of streams where the number of classes/clusters change dynamically

Mining of complex time-stamped events

Y. Matsubara, Y. Sakurai, C. Faloutsos, T. Iwata, M. Yoshikawa, 2012, 2015

Motivation

- Complex time-stamped events

  {timestamp + multiple attributes}

e.g., web click events:

{timestamp, URL, user ID, access devices, http referrer,…}

Timestamp	URL	User	Device
2012-08-01-12:00	CNN.com	Smith	iphone
2012-08-02-15:00	YouTube.com	Brown	iphone
2012-08-02-19:00	CNET.com	Smith	mac
2012-08-03-11:00	CNN.com	Johnson	ipad
…	…	…	…

Motivation

- Q: Are there any topics?
- -news, tech, media, sports, etc.?

Timestamp	URL	User	Device
2012-08-01-12:00	CNN.com	Smith	iphone
2012-08-02-15:00	YouTube.com	Brown	iphone
2012-08-02-19:00	CNET.com	Smith	mac
2012-08-03-11:00	CNN.com	Johnson	ipad
...	...	...	...

- e.g., CNN, CNET -> news, YouTube -> media

Motivation

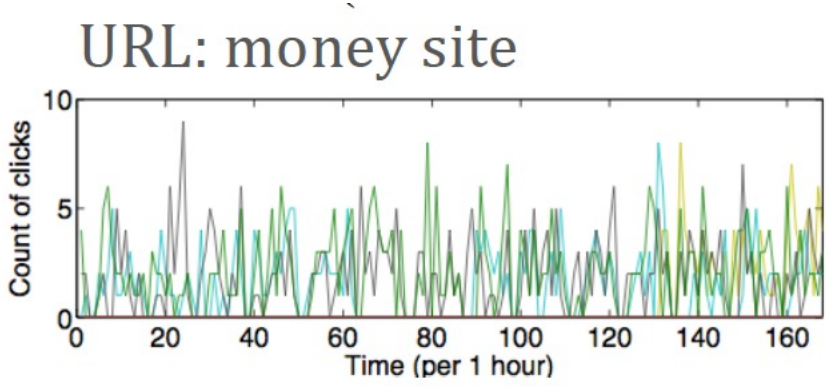
- Q: Are there any topics?
- -news, tech, media, sports, etc.?

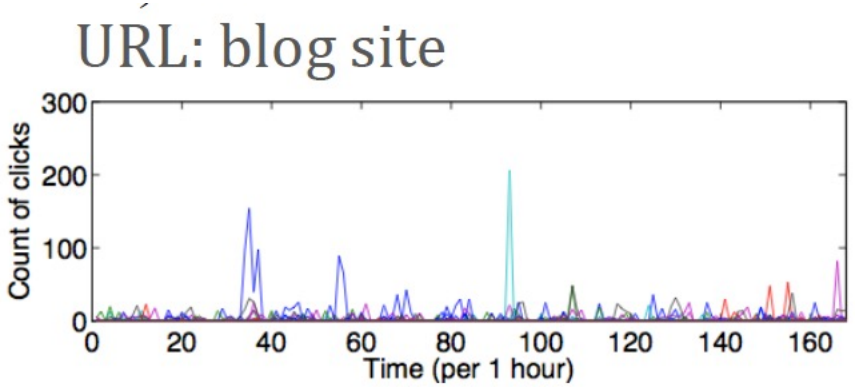
Timestamp	URL	User	Device
2012-08-01-12:00	CNN.com	Smith	iphone
2012-08-02-15:00	YouTube.com	Brown	iphone
2012-08-02-19:00	CNET.com	Smith	mac
2012-08-03-11:00	CNN.com	Johnson	ipad
...	...	...	...

- e.g., CNN & CNET (related to news)
- Smith & Johnson (related to news)

Motivation

- Web click events – can we see any trends?
- Original access counts for each URL
  - 100 random users
  - 1 week (window size 1 hour)





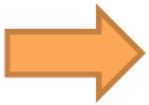
We cannot see any trend! Noisy, Sparse, Bursty

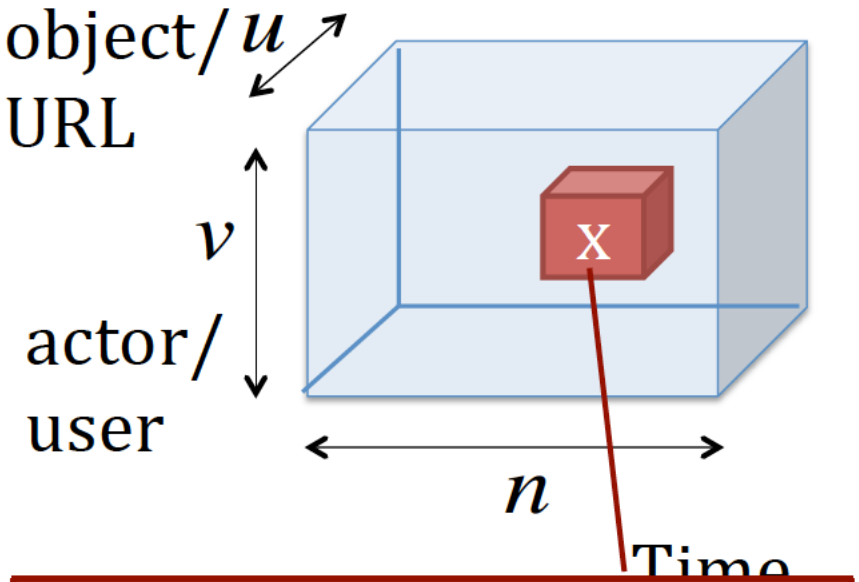
M-way analysis

- Complex time-stamped events

e.g., web clicks

Time	URL	User
08-01-12:00	CNN.com	Smith
08-02-15:00	YouTube.com	Brown
08-02-19:00	CNET.com	Smith
08-03-11:00	CNN.com	Johnson
...	...	...





Represent as
M[th] order tensor (M=3)

$$\mathcal{X} \in \mathbb{N}^{u \times v \times n}$$

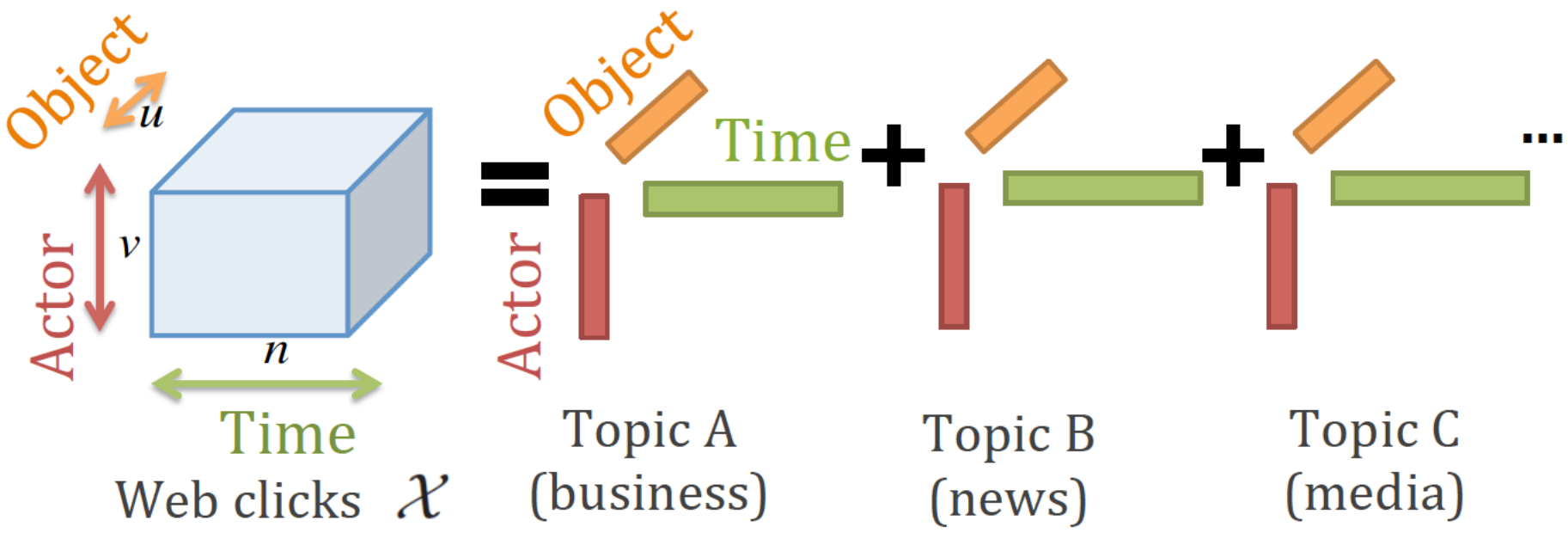
Element x: # of events

e.g., 'Smith', 'CNN.com', 'Aug 1, 10pm'; 21 times

M-way analysis

- Decompose to a set of 3 topic vectors:

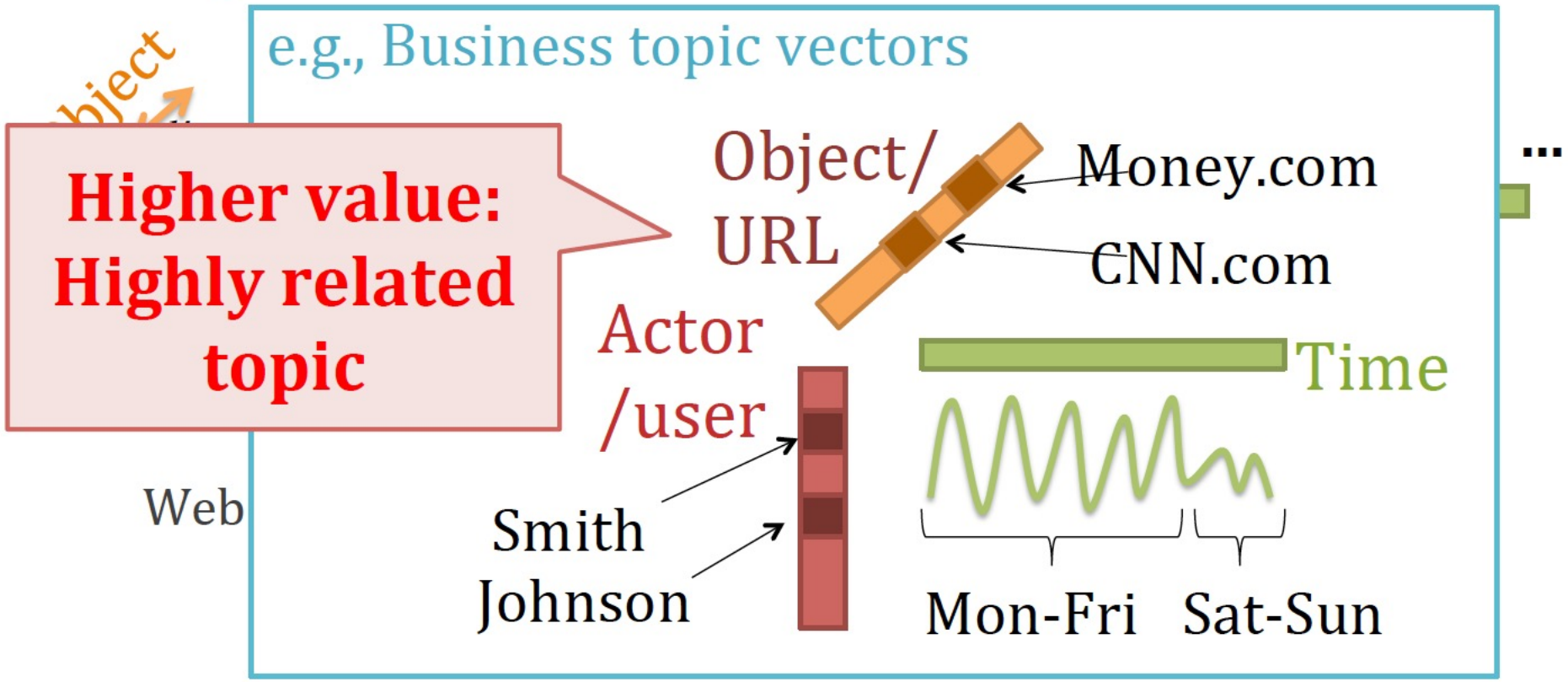
- **Object vector** **Actor vector** **Time vector**



M-way analysis

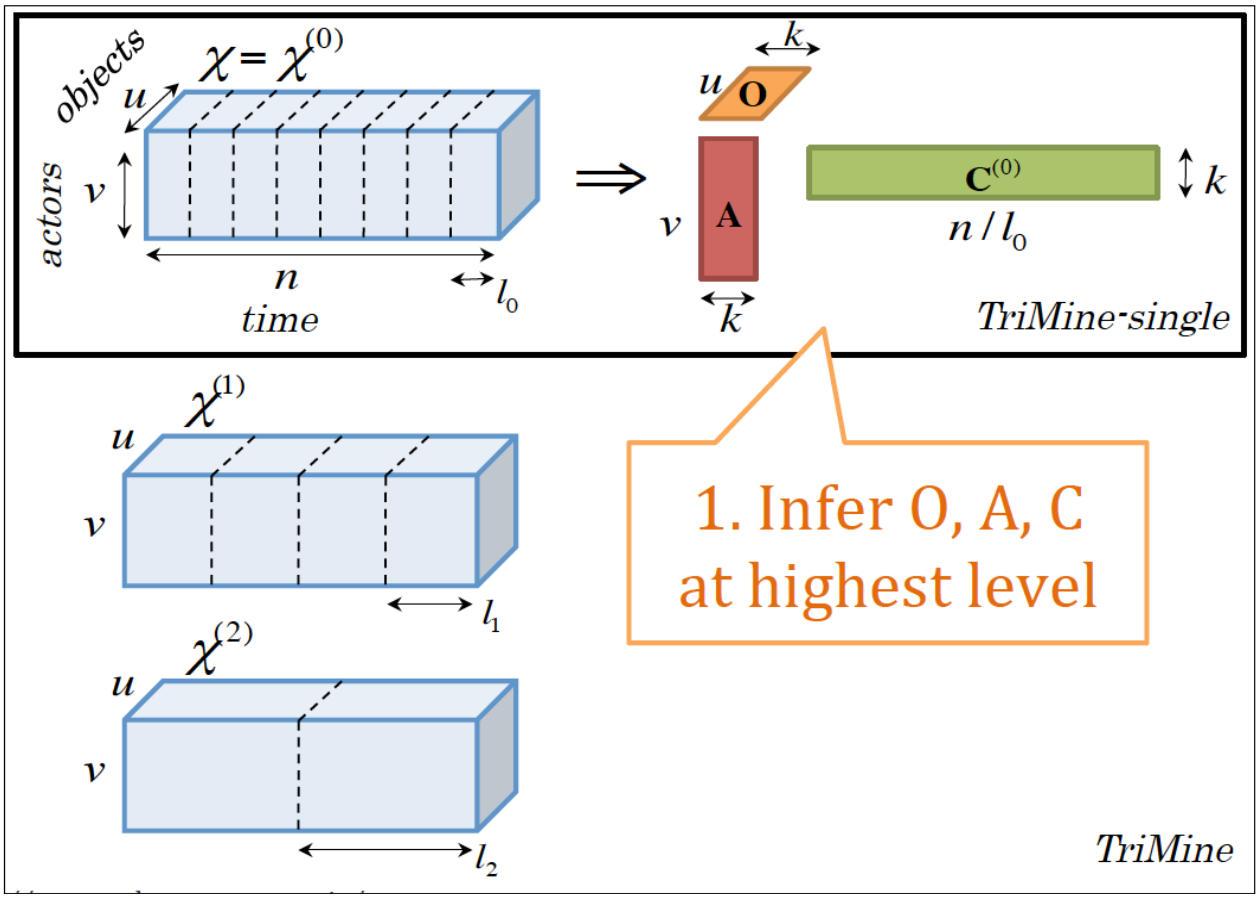
- Decompose to a set of 3 topic vectors:

- **Object vector** **Actor vector** **Time vector**



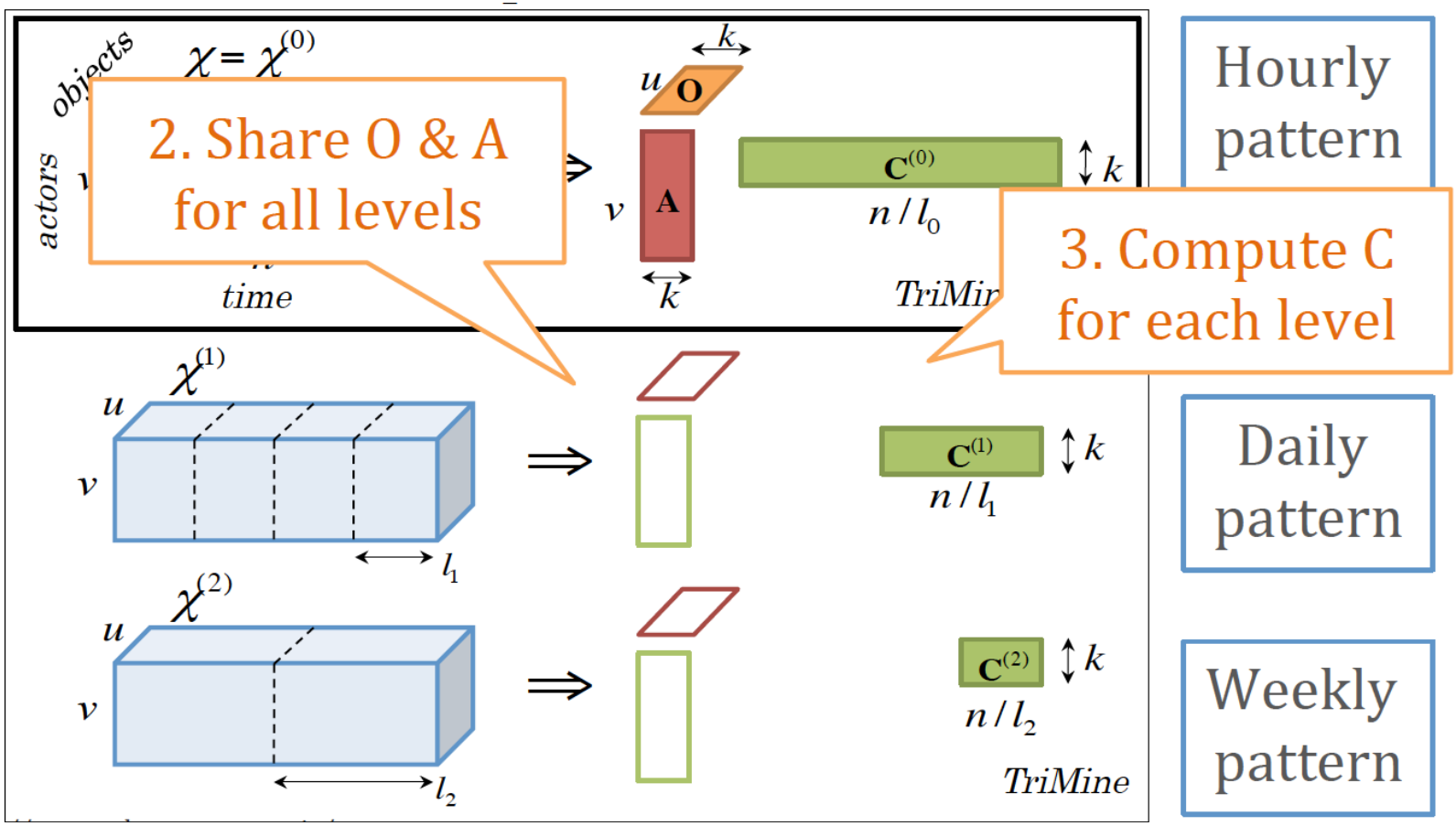
M-way analysis

- Tensors with multiple window sizes:



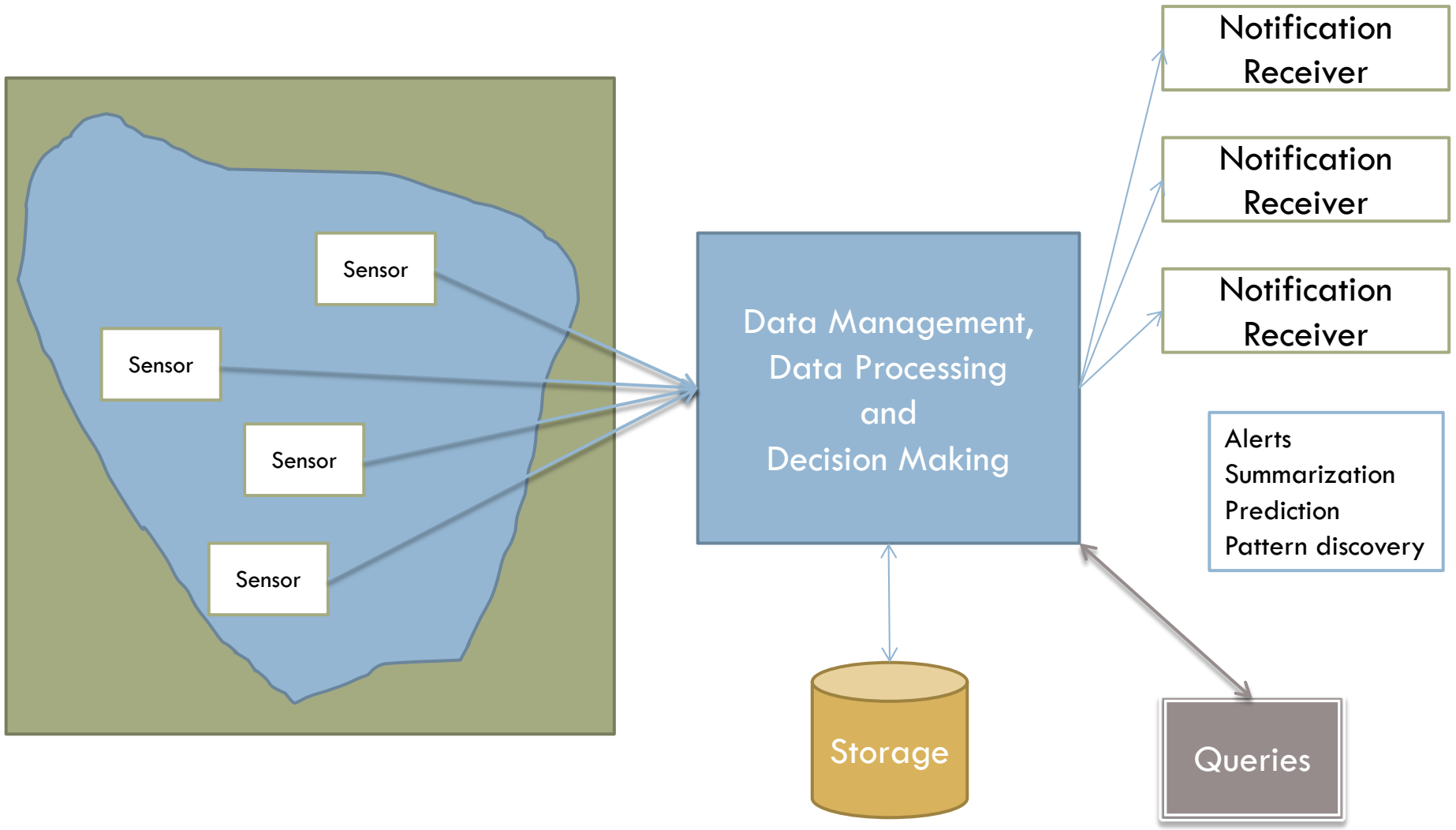
M-way analysis

- Tensors with multiple window sizes:



Data Stream Management

Data Stream Management



Data stream characteristics and requirements

- High data transfer rates
- Transfer Rates different for each sensor
- Fast increase of volume of data

Basic requirements in data stream management:

- Efficient processing of recent data
- Efficient retrieval of past data
- Efficient storage/compression of data

Data stream management goal

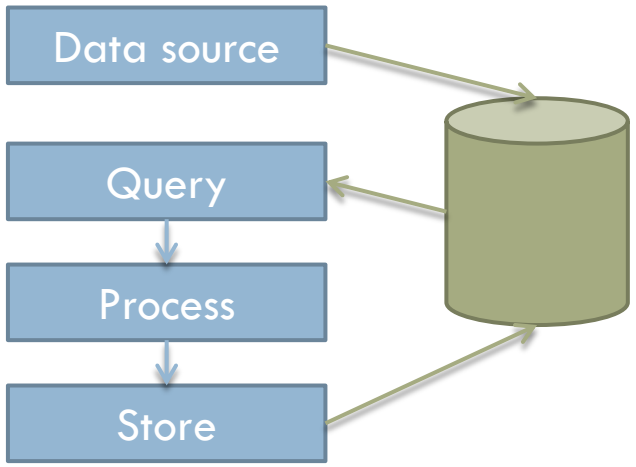
Efficient support of the data analytics algorithms

Use of algorithms with the following characteristics:

- Low complexity
- As high accuracy as possible
- Utilization of information from various sources of data considering parallel evolution

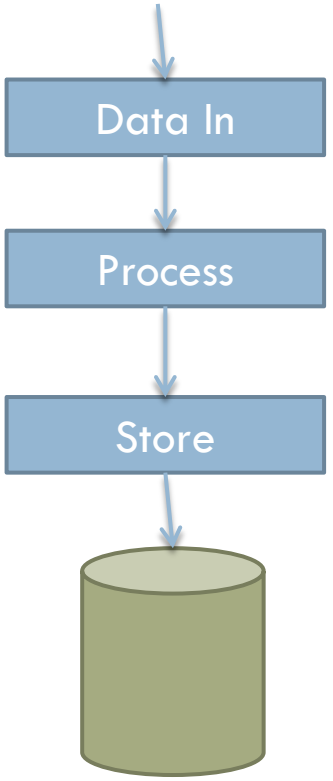
Data Stream Management

- Why not traditional DBMSs;
  - Goal of DBMSs: data storage for applications consisting of the followings four steps:
    1. Data storage
    2. Data retrieval
    3. Data Processing
    4. Data Storage



Data Stream Management

- Instead, typical applications on data streams consists of the following steps:
  1. Data entering the application
  2. Data processing
  3. Data storage



Processing of data usually occurs before storage

Data stream management – Queries

- Many data stream applications require the execution of SQL queries
- The queries are executed on data that are updated continuously
- Key features / types of Queries:
  - Time Based Queries
  - **Continuous Queries**
  - Short Term Queries
  - Long Term Queries

Classic DBMSs do not support these operations.. or rather they are not optimized for these operations!

Databases – Queries Static vs Continuous

Static Query

- Return the number of students with GPA above 8.5/10.0

Continuous Query

- Return the mean value of the pollution indicator (+ time constraints)

In addition:

- The output of a continuous query can be a new stream
- The continuous queries can consist of subqueries as in traditional DBs

Continuous Query -Example

The following query (Linq) is being executed continuously and produces a new stream

```
var polutionSensorValue = from e in sensorstream
                          where e.id == 312343
                          select e.value;
```

Issues in data stream management

Some additional problems..

- When query starts, not all data may be available
- The size of a stream is unlimited
- The frequencies/rates at which data from different streams arrive are not the same
- It is not clear:
  - When is the output of a query ready?
  - In what order do the data arrive?
  - How long should we wait for new data?

Management Systems
Data Base vs Data Stream

DBMS

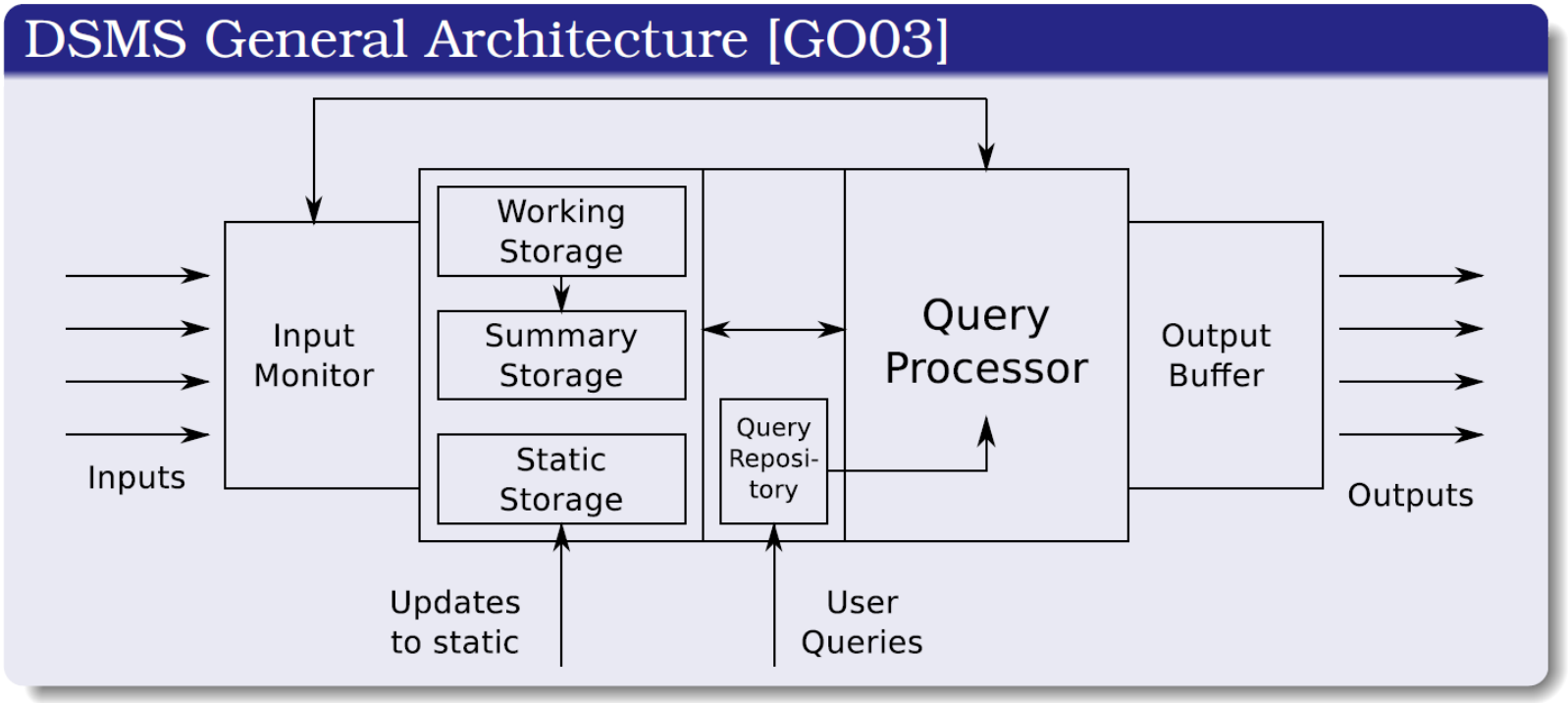
- Static Data
- On-Demand execution of queries
- Exact results
- Disk storage

DSMS

- Continuous Data
- Continuous Execution of Queries
- Results: Usually Approximate
- Storage in main memory

DSMS: Data Stream Management Systems

- DSMS General Architecture [GO03]



DSMS - Queries
Blocking Operators

Problem

The operations:

- Sort
- Join
- Group

are blocking…

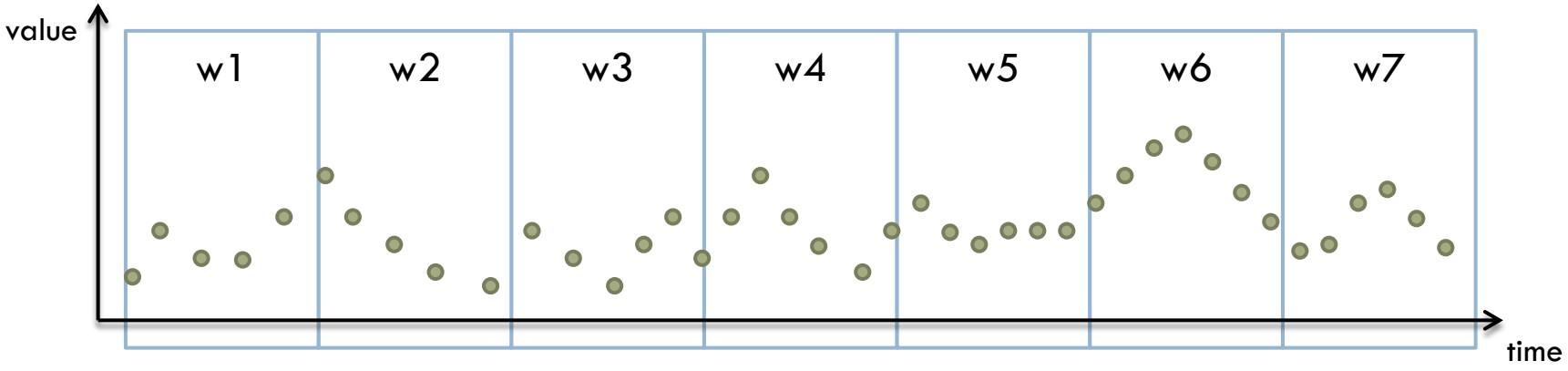
Solutions

- Random Subsets
- Fixed and Sliding Windows
- Punctuation

DSMS – Windows

Hopping Window

- In most applications that process data streams, calculations are performed on data that involve continuous time intervals (e.g. calculation of the lake level every 5 minutes)
- Basic function of DSMSs: the ability to **process data in time windows** defined by developers



Basic Parameters:

- Window Size
- Hop Size

Types of windows:

- Time-based (e.g., 15 secs)
- Count-based (e.g., 50 samples)

DSMS - Queries

Query Scheduler

- Στο “Query Repository” υπάρχουν τα queries που έχουν τεθεί
- Βασικό πρόβλημα είναι η σειρά με την οποία θα τρέξουν τα ερωτήματα αυτά
- Optimization Πρόβλημα:
  - Ελάχιστος χρόνος εκτέλεσης
  - Ελάχιστη μνήμη

DSMS - Queries

Query Scheduler, Load Shedding

Four dimensions of the problem [SLC08]

- Setting goals and implementing policies for their implementation (eg, minimum latency, less memory)
- Window-Based CQs - launch complex queries such as joins
- Performance optimization utilizing common sub-queries
- Scheduler implementation in an optimal way - necessary for online systems

When system is overloaded and not all data can be processed, a DSMS must provide a strategy to "cut" some of the load with the least loss of accuracy (Tatbul et al., 2003] and [Babcock et al. ., 2004]).

DSMS - Storage

- Good organization of the historic data necessary for long term queries
  - mainly for data warehouses
- Classic methods such as B-Trees support many data types
- For time series?
- Temporary Storage?

DSMS - Systems and Query Languages

- Aqsios [SCLP08]
- AQuery [LS03]
- Aurora [AC+03]
- CQL/STREAM [ABW06]
- XStream [GMN+08]
- DataDepot [GJSS09]
- StreaQuel
- Tribeca

They differ mainly in:

- applications in which they are oriented
- the operators they support

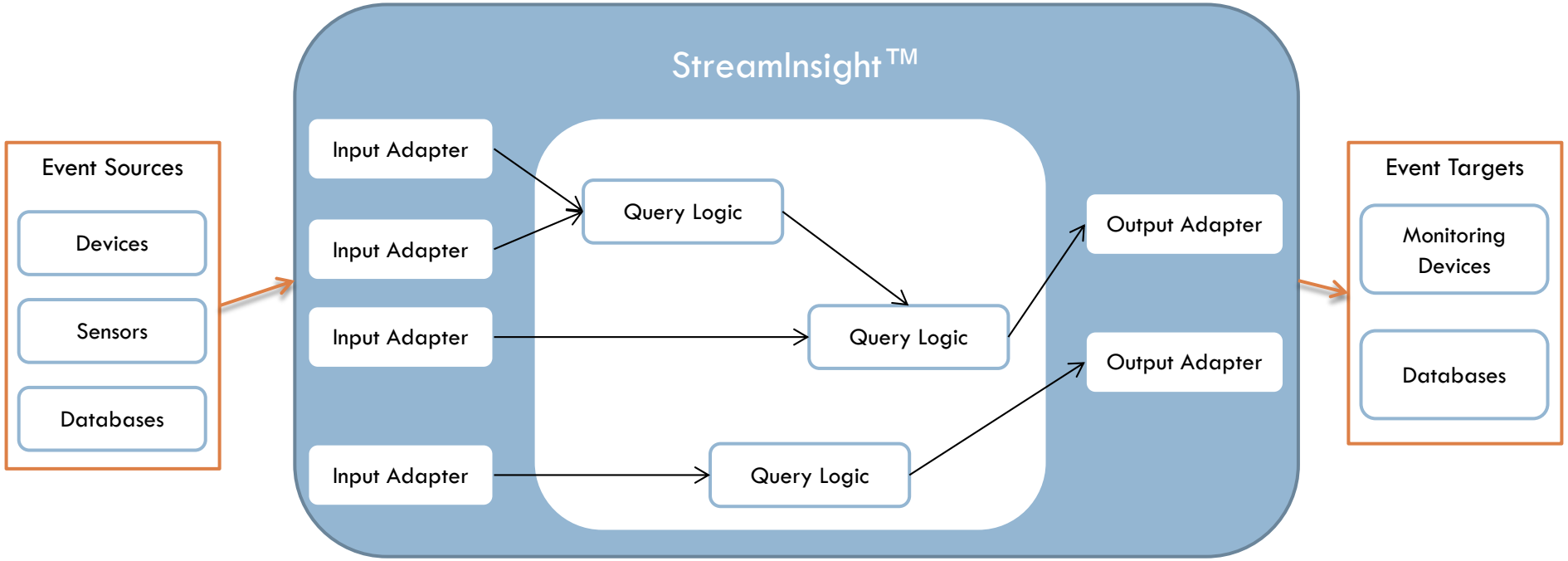
DSMS - Systems and Query Languages

Language/ system	Motivating applications	Allowed inputs	Basic operators	Supported windows			Custom operators?
				type	base	execution	
AQuery	stock quotes, network traffic analysis	sorted relations	relational, "each", order-dependent (first, next, etc.)	fixed, landmark, sliding,	time and count	not discussed in [65]	via "each" operator
Aurora	sensor data	streams only	σ,π,$\cup$,$\bowtie$, group-by, resample, drop, map, window sort	fixed, landmark, sliding	time and count	streaming	via map operator
CQL/ STREAM	all-purpose	streams and relations	relational, relation-to-stream, sample	currently only sliding	time and count	streaming	allowed
StreaQuel/ TelegraphCQ	sensor data	streams and relations	relational	all types	time and count	streaming or periodic	allowed
Tribeca	network traffic analysis	single input stream	σ, π, group-by, union aggregates	fixed, landmark, sliding	time and count	streaming	allows custom aggregates

Example DSMS: Microsoft StreamInsight

- Data Inputs:
  - Input/Output Adapters
  - Observable and Enumerable sources/sinks.
- Processing within StreamInsight™ is commonly performed by writing queries in Linq (Language Integrated Query)
- User Defined Operators (UDOs), User Defined Functions (UDFs) and User Defined Aggregates (UDAs)
- StreamInsight™ offers different types of windows, such as hopping, count and snapshot

Microsoft StreamInsight™



XStream DSMS [GMN+08]

- Signal Oriented DSMS
- Basic characteristics
  - WaveScript programming language
  - SigSeg data type
  - Unified Query Language
  - Memory Management
  - Query Plans
  - Query Optimization
  - High Performance
  - Distributed Execution Engine

References

- Abadi, D. J., Carney, D., Hetintemel, U., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N., and Zdonik, S. (2003). Aurora: a new model and architecture for data stream management. The VLDB Journal, 12:120–139.
- Arasu, A., Babcock, B., Babu, S., Cieslewicz, J., Datar, M., Ito, K., Motwani, R., Srivastava, U., and Widom, J. (2004). Stream: The stanford data stream management system. Technical Report 2004-20, Stanford InfoLab.
- Babcock, B., Babu, S., Datar, M., Motwani, R., and Widom, J. (2002). Models and issues in data stream systems. In Proceedings of the twenty-first ACM SIGMODSIGACT- SIGART symposium on Principles of database systems, PODS '02, pages 1–16, New York, NY, USA. ACM.
- Babcock, B., Babu, S., Motwani, R., and Datar, M. (2003). Chain: operator scheduling for memory minimization in data stream systems. In Proceedings of the 2003 ACM SIGMOD international conference on Management of data, SIGMOD '03, pages 253–264, New York, NY, USA. ACM.
- Babcock, B., Datar, M., and Motwani, R. (2004). Load shedding for aggregation queries over data streams. Data Engineering, International Conference on, 0:350.

References

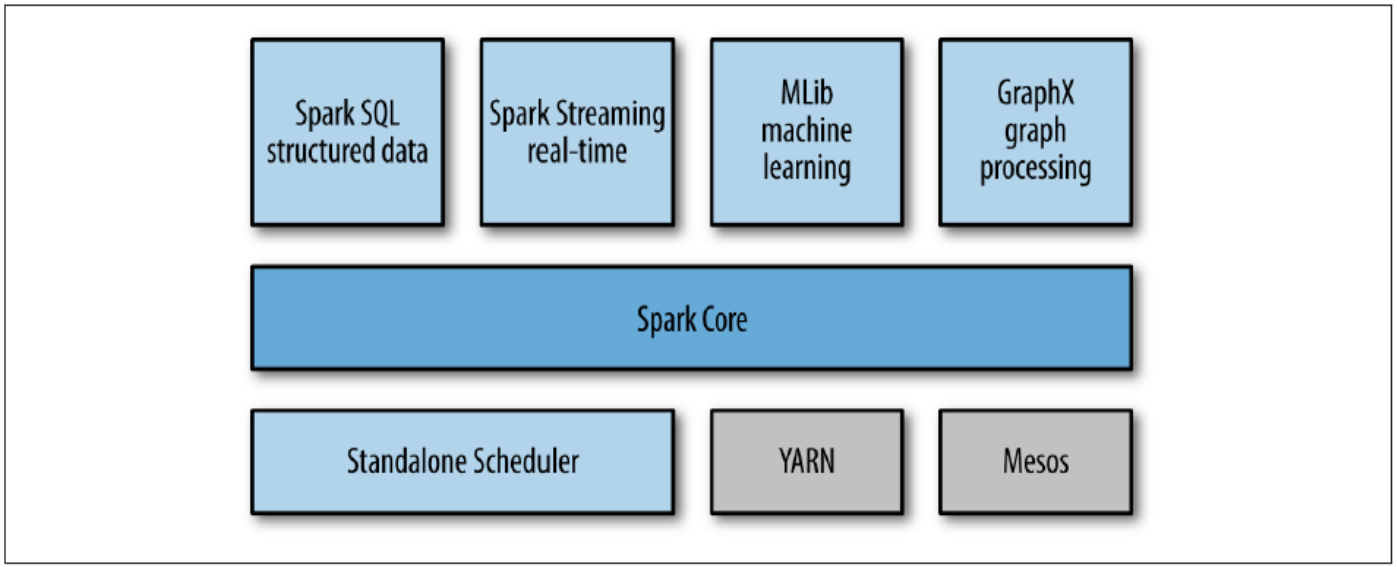
- Girod, L., Mei, Y., Newton, R., Rost, S., Thiagarajan, A., Balakrishnan, H., and Madden, S. (2008). Xstream: a signal-oriented data stream management system. In Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on, pages 1180 –1189.
- Golab, L. and Fzsu, M. T. (2003). Issues in data stream management. SIGMOD Rec., 32:5–14.
- Sharaf, M. A., Labrinidis, A., and Chrysanthis, P. K. (2008).Scheduling continuous queries in data stream management systems. Proc. VLDB Endow., 1:1526–1527.
- Tatbul, N., Hetintemel, U., Zdonik, S., Cherniack, M., and Stonebraker, M. (2003). Load shedding in a data stream manager. In Proceedings of the 29th international conference on Very large data bases - Volume 29, VLDB '2003, pages 309–320. VLDB Endowment.
- Zhu, Y. and Shasha, D. (2002). Statstream: statistical monitoring of thousands of data streams in real time. In Proceedings of the 28th international conference on Very Large Data Bases, VLDB '02, pages 358–369. VLDB Endowment.


APACHE
Spark

Spark-Architecture

- ***Unified pile*** → **Multiple correlated levels**
- ***Spark Core*** → basic functionality → scheduling of tasks, memory management, failure recovery, etc
- ***Spark SQL*** → management and storage of structured data
- ***Spark Streaming*** → management of streams in real time
- ***Mllib*** → libraries for Machine Learning
- ***GraphX*** → graph management
- ***Cluster Managers*** → efficient scaling from one to hundreds of computing nodes (Standalone Scheduler, Hadoop YARN, Apache Mesos)

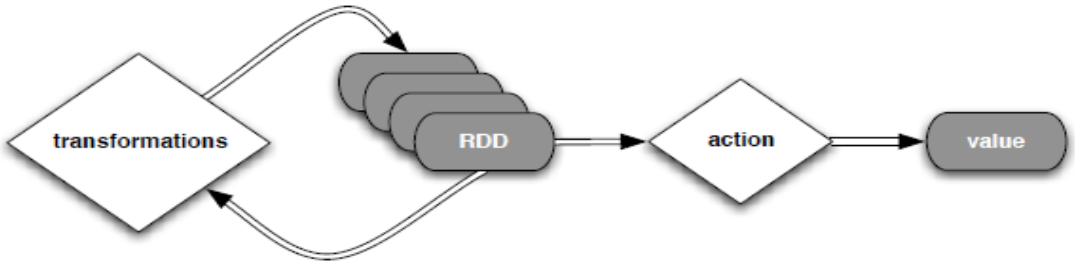
- **Creating applications that combine different processing models**



Spark- programming model

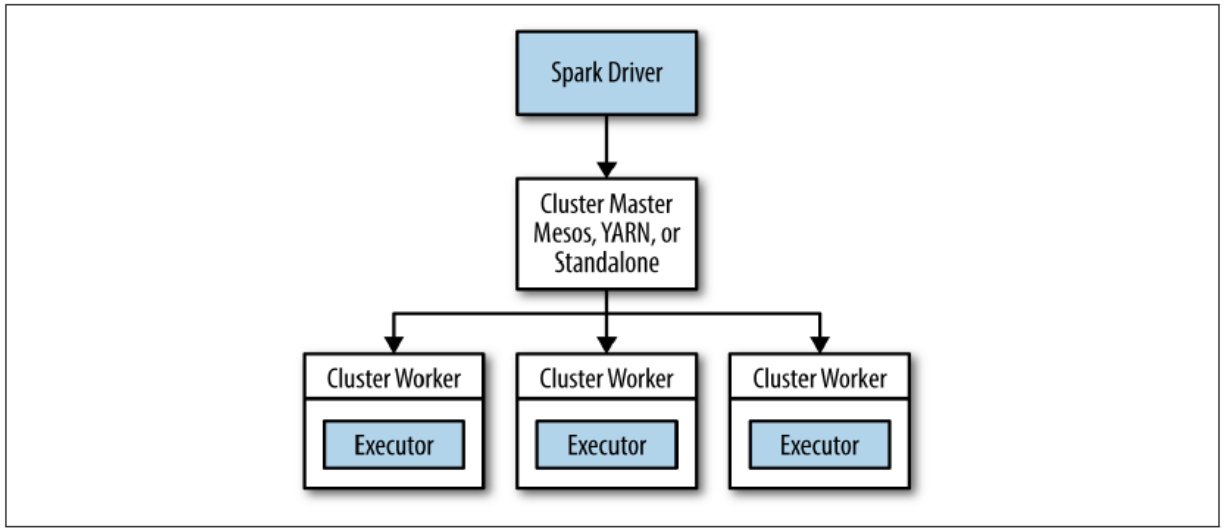
- Main abstract programming entity → **RDDs (*resilient distributed data-sets*)** → distributed set of objects that are in different computational nodes
- Basic functionalities of RDDs
  - **Transformations** (lazy) → construct new RDDs from existing
  - **Actions** → compute results based on the running RDD

➢ **Basically, RDDs give us the capability of parallel execution**



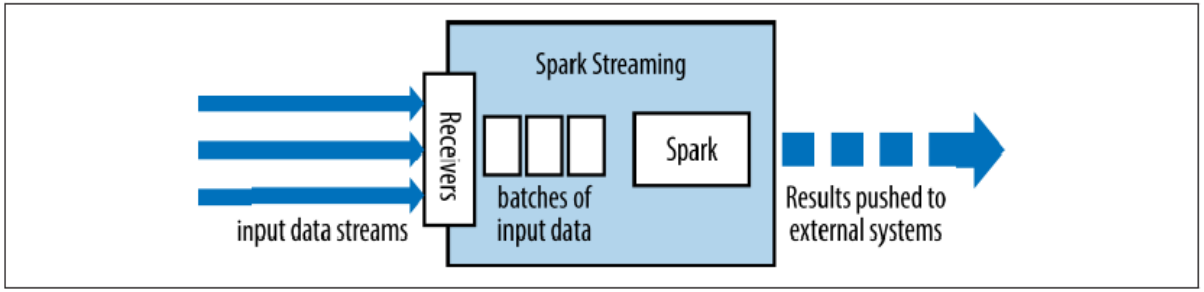
Spark- model of execution

- ***Distributed execution → master/slave architecture***
- **Spark driver** → central coordinator, communicates with the distributed workers (Executors)
  - Conversion of user program to Tasks
  - Scheduling of Tasks to Executors
- **Executors** → worker processes, perform tasks, memory
- **Spark driver and Executors** → Spark application → starts on a set of engines using an external service *"cluster manager"*

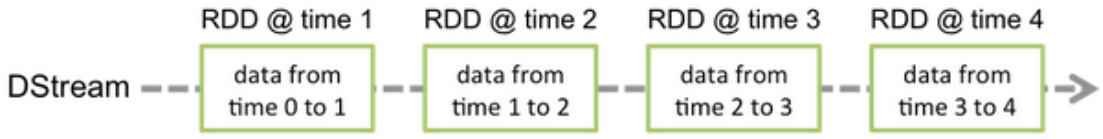


Spark Streaming API

- Management of data streams in real time
- **"micro batch" architecture** → data from input sources are organized in batches
- As time passes new batches are generated in regular time intervals → each batch is essentially an RDD
- The smaller the batch (time wise) the more we approximate the Streaming logic



➢ Abstract programming entity of Spark Streaming → **discretized stream (DStream)** = a sequence of RDDs, each RDD contains a segment of time from data stream



Putting streaming data management and analytics to work

Advanced multi-paRametric Monitoring and analysis for diagnosis and Optimal management of epilepsy and Related brain disorders

Dept. of Computer Engineering and Informatics
University of Patras
Rion-Patras, Greece

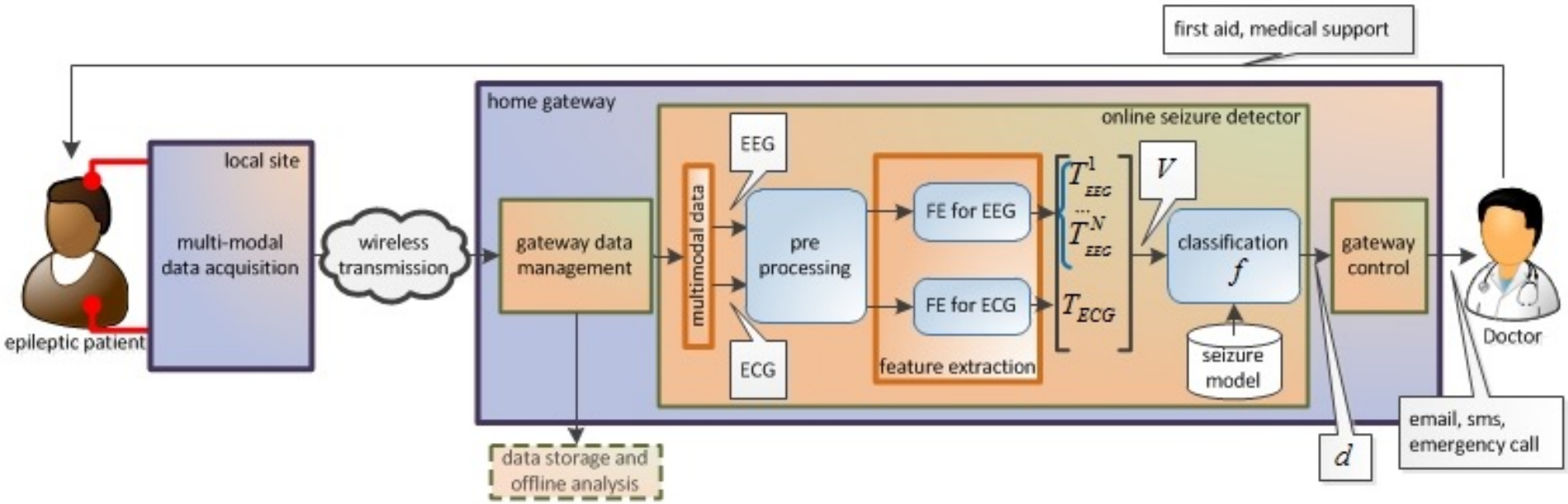
Dept. of Clinical Neurophysiology and Epilepsies
Guy's & St. Thomas' and Evelina Hospital for Children
NHS Foundation Trust, London, UK

D. Triantafyllopoulos, P. Korvesis, I. Mporas and V. Megalooikonomou, "Real-Time Management of Multimodal Streaming Data for Monitoring of Epileptic patients", Journal of Medical Systems, 40(3), pp. 1-11, 2016.

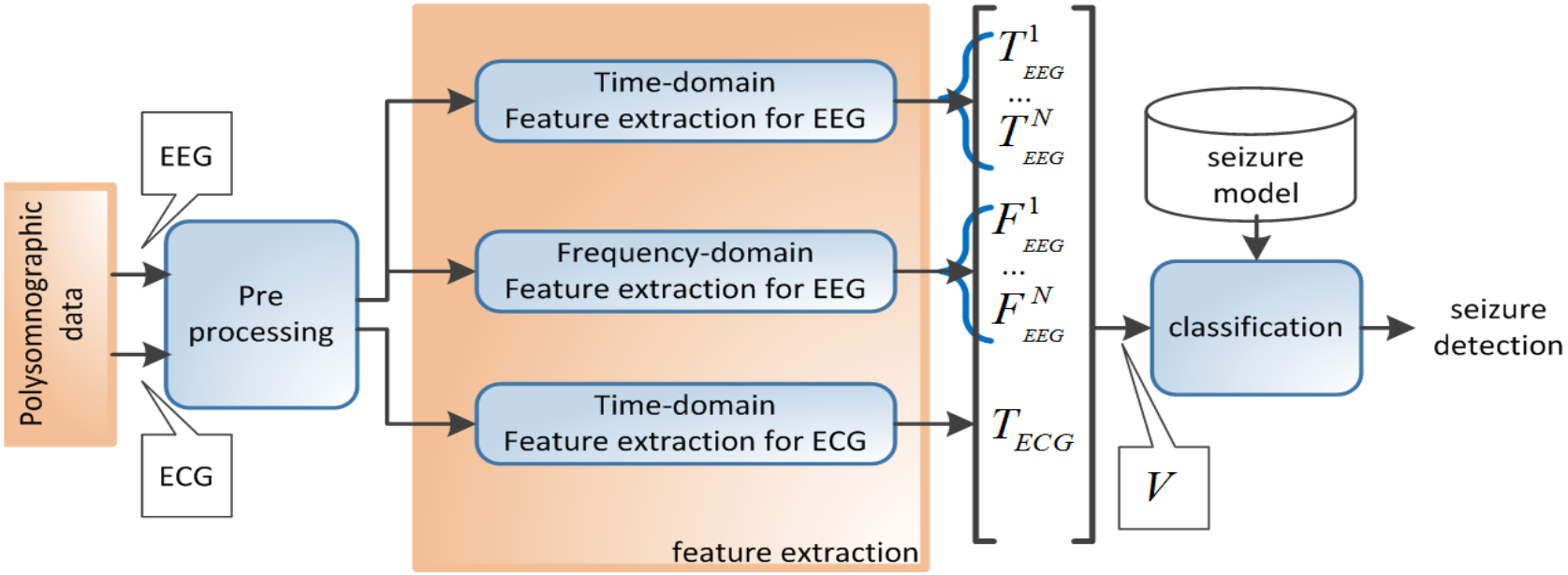
I. Mporas, V. Tsirka, E. I. Zacharaki, M. Koutroumanidis, M. Richardson, V. Megalooikonomou, "Seizure detection using EEG and ECG signals for computer-based monitoring, analysis and management of epileptic patients", Expert Systems with Applications, 42, pp. 3227-3233, 2015.

European Commission
Information Society and Media
SEVENTH FRAMEWORK PROGRAMME

Online Seizure Detection concept



Online Seizure Detection



- Based on EEG, ECG data
- Seizure Model: binary SVM-model
  - subject-specific model
  - trained off-line

Online Seizure Detection: features used

- EEG data
  - Time domain features
    - min, max, mean, variance, std, percentiles, interquartiles
    - range, skewness, kyrtosis, energy, zero-crossing rate
    - Shannon entropy, log-energy entropy
  - Frequency domain features
    - AR-coefficients (6)
    - power spectral density
    - max/min frequency
    - {δ , θ , α, β, γ} band energy
    - 8 DWT-based band energies (daubechies-16)
- ECG data
  - RR statistics (HRV): min, max, mean, standard deviation, variance, percentiles, interquartile range, mean absolute deviation, range

Online Seizure Detection

- Performance
  - Data
    - CHB-MIT Scalp EEG+ECG Database
    - Subject: 04
  - Detection Accuracy (96.31 %)

Classified as →	Seizure	Not
Seizure	94.74 %	05.26 %
Not	03.68 %	96.32 %

Online Seizure Detection

- Performance
  - Data
    - St. Thomas recordings EEG+ECG, Subjects: 07, 08, 09
  - Subject 07: Accuracy (99.85%)

Classified as →	Seizure	Not
Seizure	92.31	07.69
Not	0.09	99.91

  - Subject 08: Accuracy (99.79%)

Classified as →	Seizure	Not
Seizure	77.78	22.22
Not	00.16	99.84

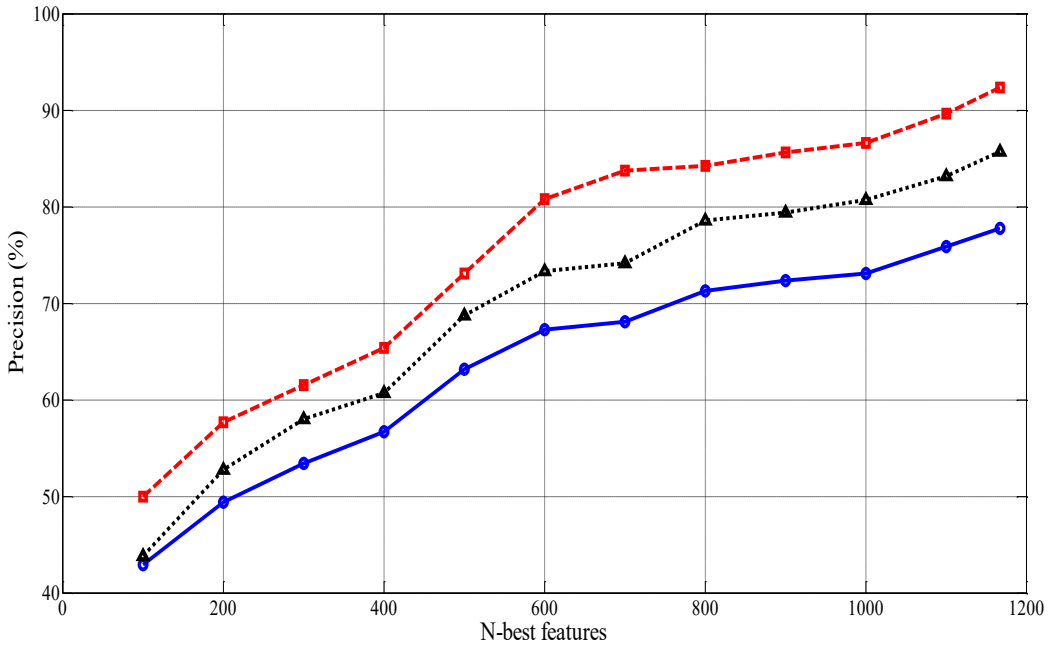
  - Subject 09: Accuracy (99.13%)

Classified as →	Seizure	Not
Seizure	85.71	14.29
Not	00.76	99.24

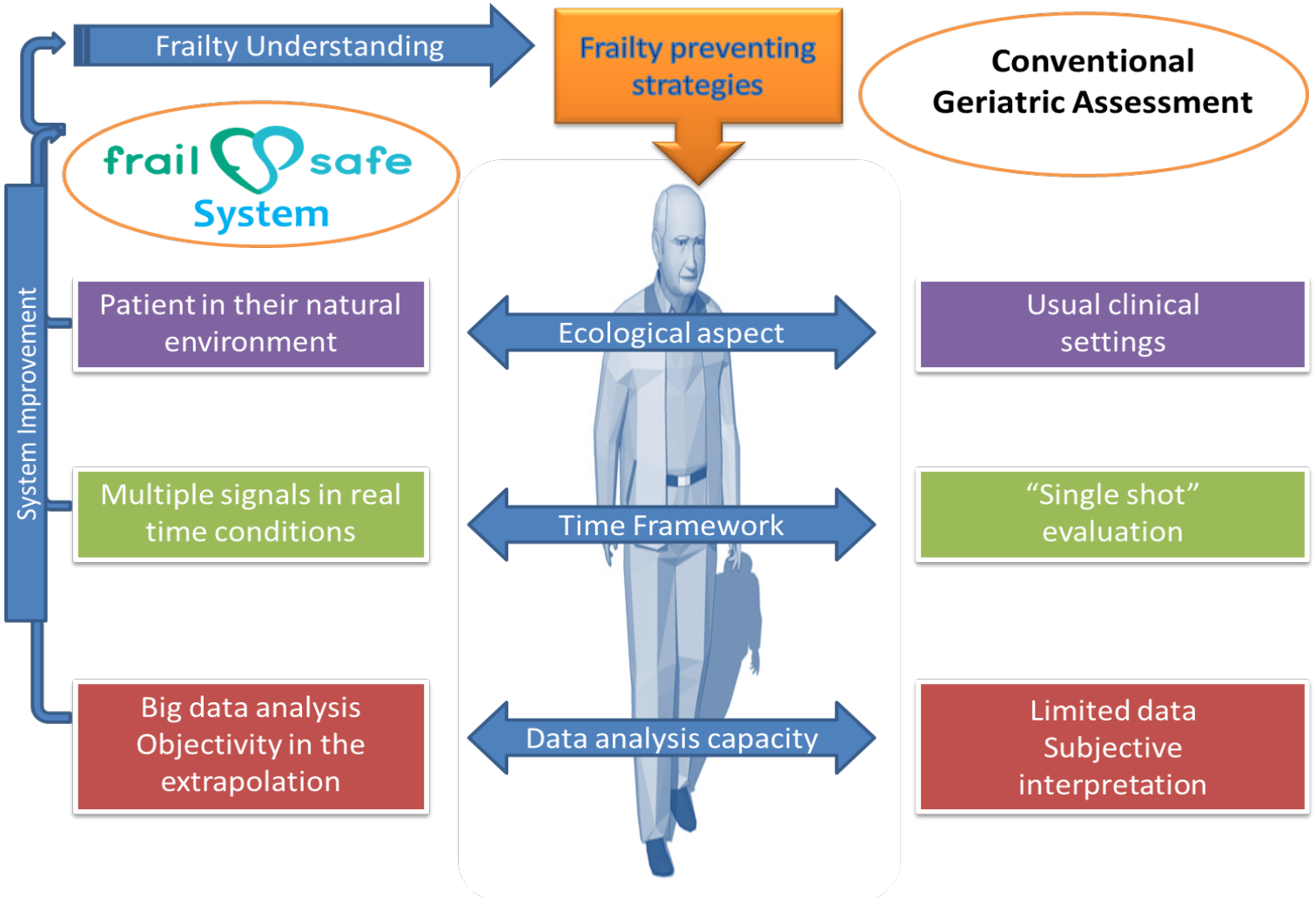
Online Seizure Detection

- Performance
  - Data: St. Thomas recordings EEG+ECG, Subjects: 07, 08, 09

Subject	SVM	MLP	C4.5	IBk
sub-07	96.11	92.93	81.91	91.67
sub-08	88.81	82.37	76.02	80.61
sub-09	92.48	87.50	79.34	84.86



Frailsafe vs Conventional Geriatric Assessment



FrailSafe offers hi-tech, clinically usable tools that lead to an earlier identification of frailty or pre-frail conditions, and makes feasible the application of early interventions to prevent worsening or reverse this condition

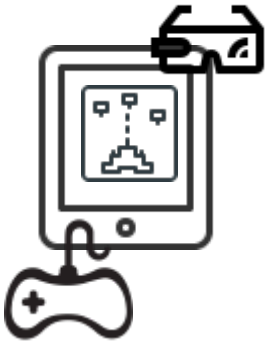
Frailsafe

- A **real life sensing** (physical, cognitive, psychological, social) **platform**
- Better understanding of frailty and its relation to co-morbidities
- **Quantitative and qualitative measures of frailty** (through advanced data mining approaches on multiparametric data)
- Prediction of short and long-term outcome and risk of frailty
- An **intervention** (guidelines, real-time feedback, AR serious games) **platform** offering personalized physiological reserve and external challenges
- A **safe, unobtrusive and acceptable system** for the ageing population

System features for the older person

indoor and outdoor monitoring

Older person



mobile and augmented reality games

smart garment

Dashboard

Third parties' devices

Measurable parameters and units of measurement

 Sensorized vest/strap with 9 DoF IMUs	Heart rate, respiration rate, posture and/or activity, steps/minute, falls, instability
 Smartphone	Indoor/Outdoor activities, Physiological state, Motor state, Social Interaction
 Questionnaires	Nutrition, Social Interaction, Cognitive state
 Medical record	Co-morbidities, etc
 Smart home sensors	Indoor activities, ….
 Dynamometer	Grip strength
 AR Serious Game	Cognitive state and Behaviour, Physiological state, Motor state
 Impedance scale	Body Weight / Body Mass Index
 Blood pressure monitor	Blood pressure
 Mobil-o-graph	Arterial stiffness

FrailSafe Conceptual Philosophy

Older Person

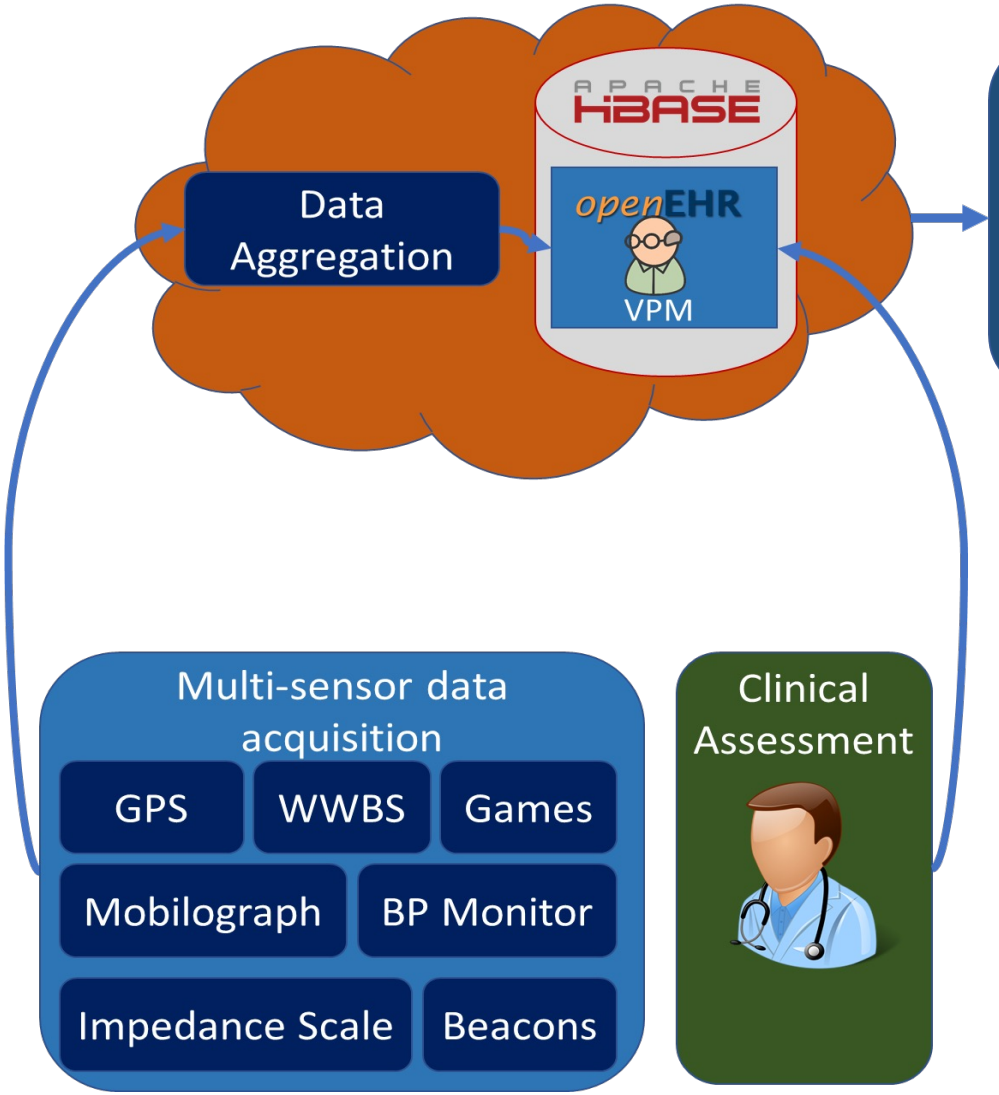
Guidelines

Virtual Patient Model (VPM)

Monitors
Designs Interventions (adjustment of drugs/drug dosage, lifestyle recommendations)

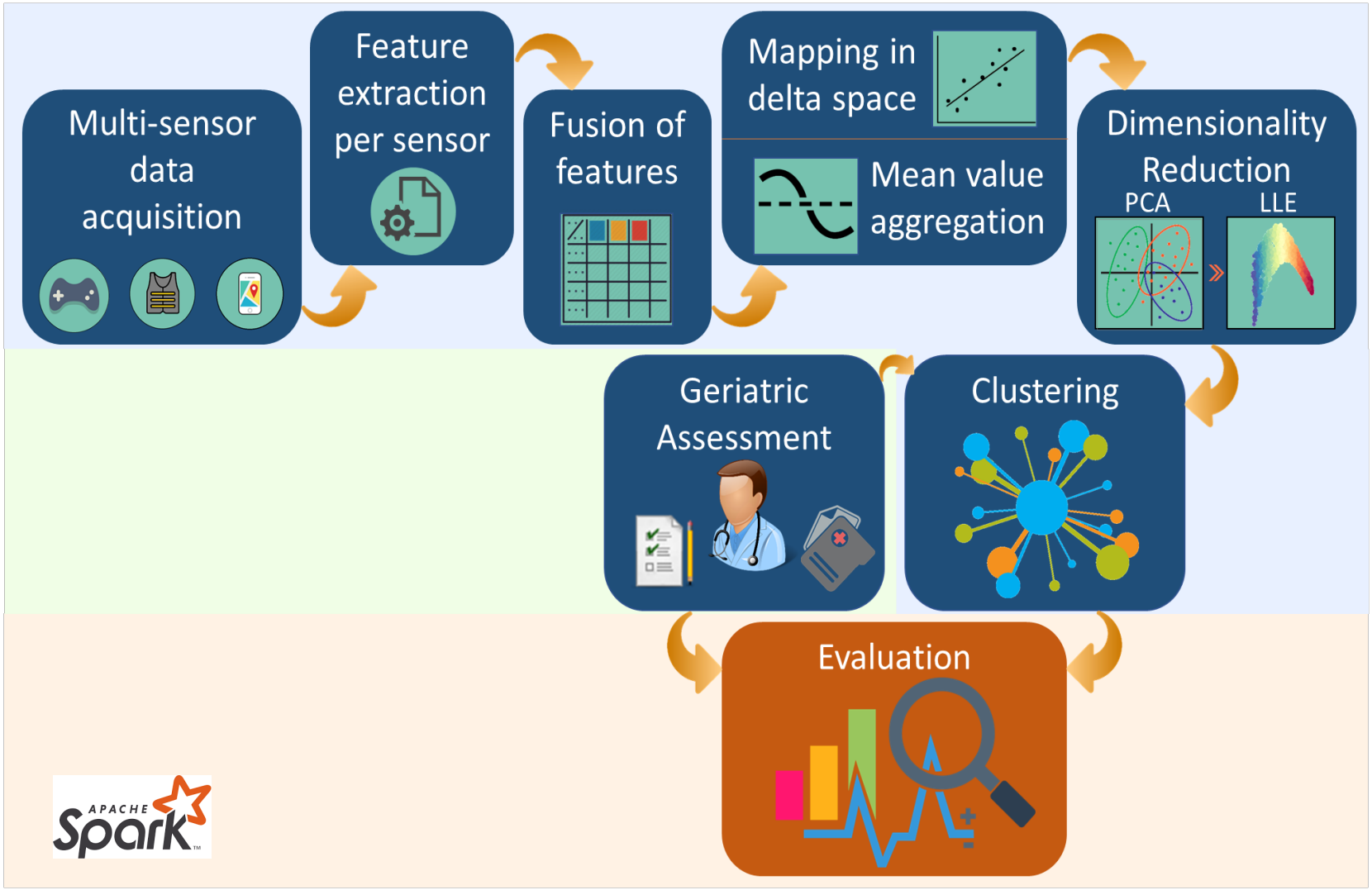
Clinician

Data Acquisition and Management

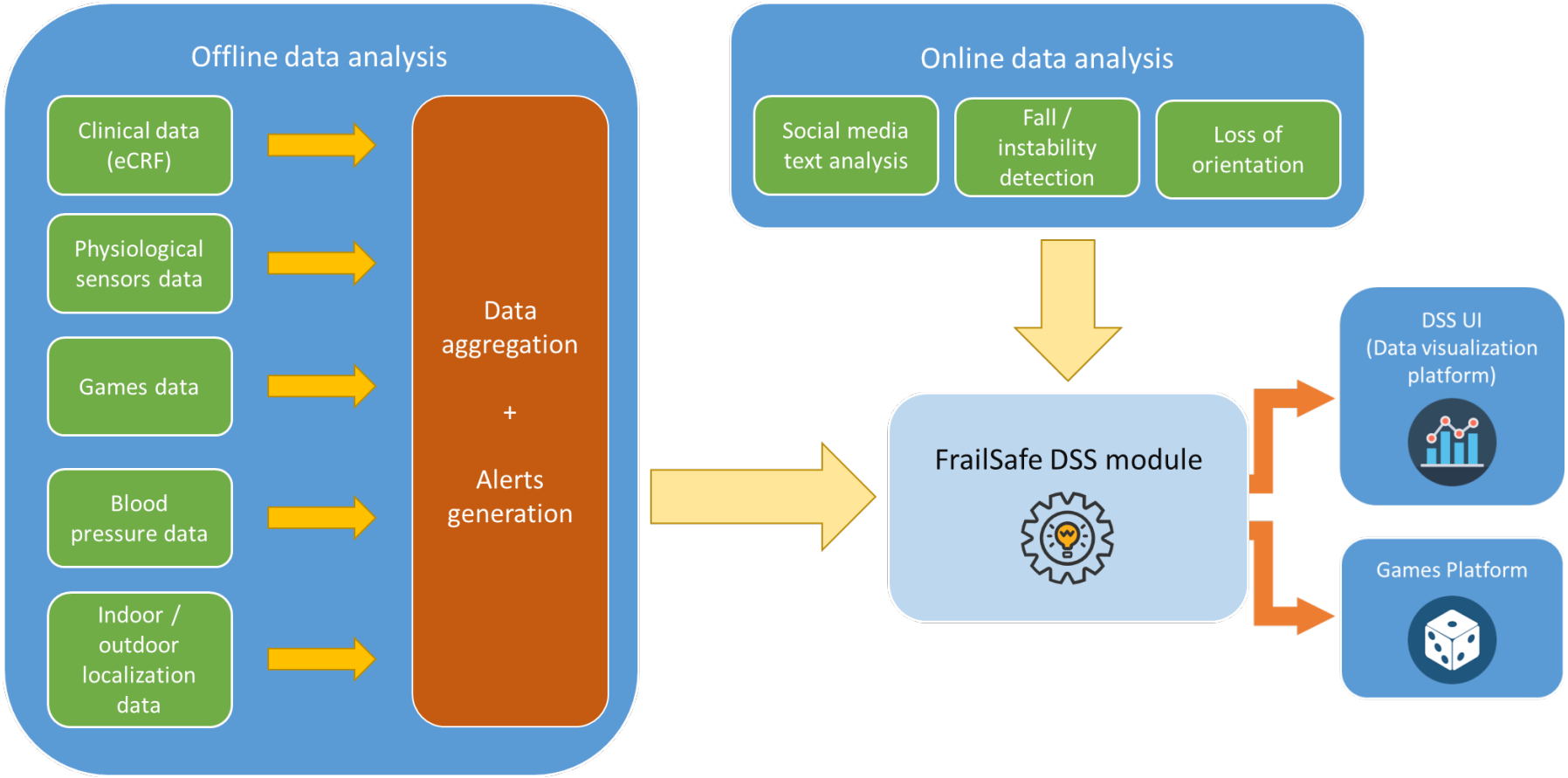


400+ participants (aged 70+)

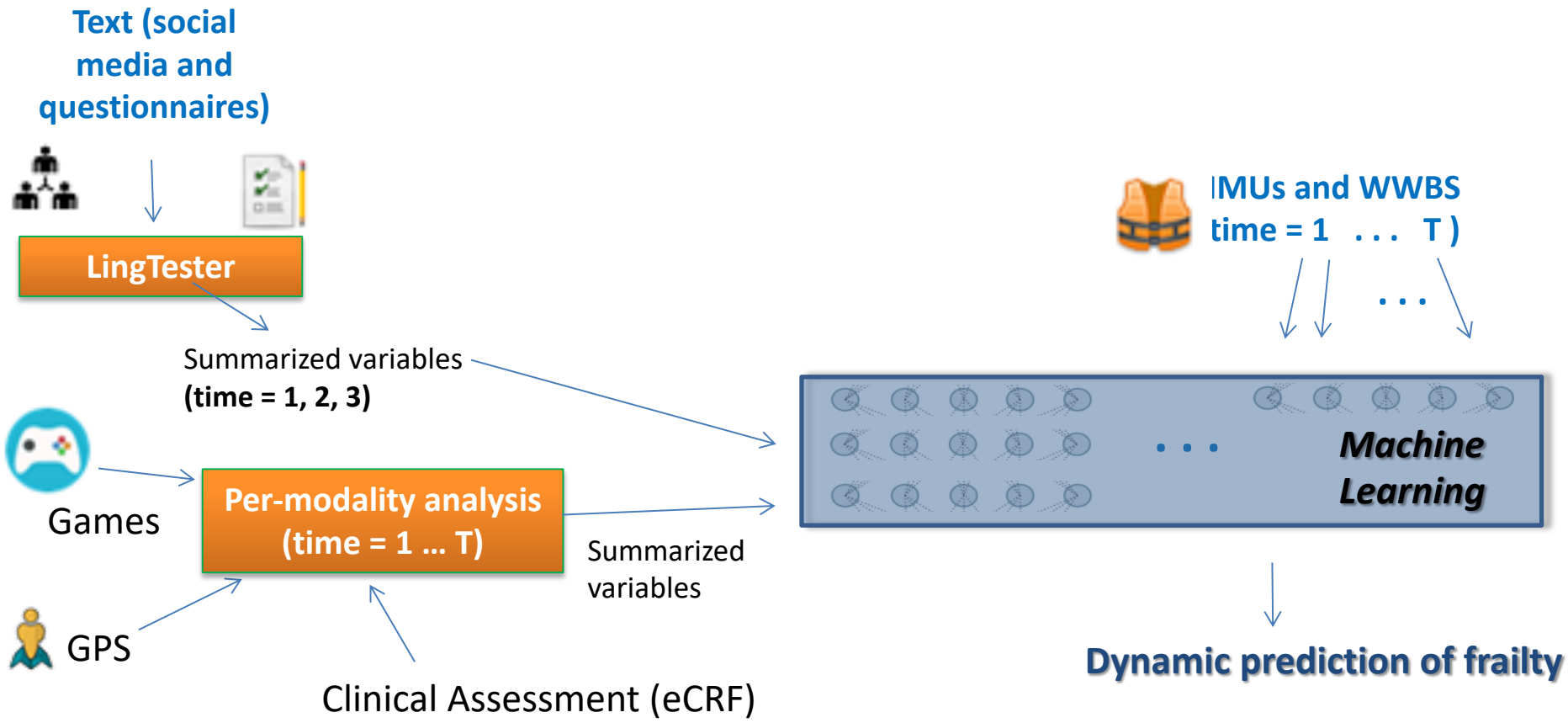
Big Data Analytics



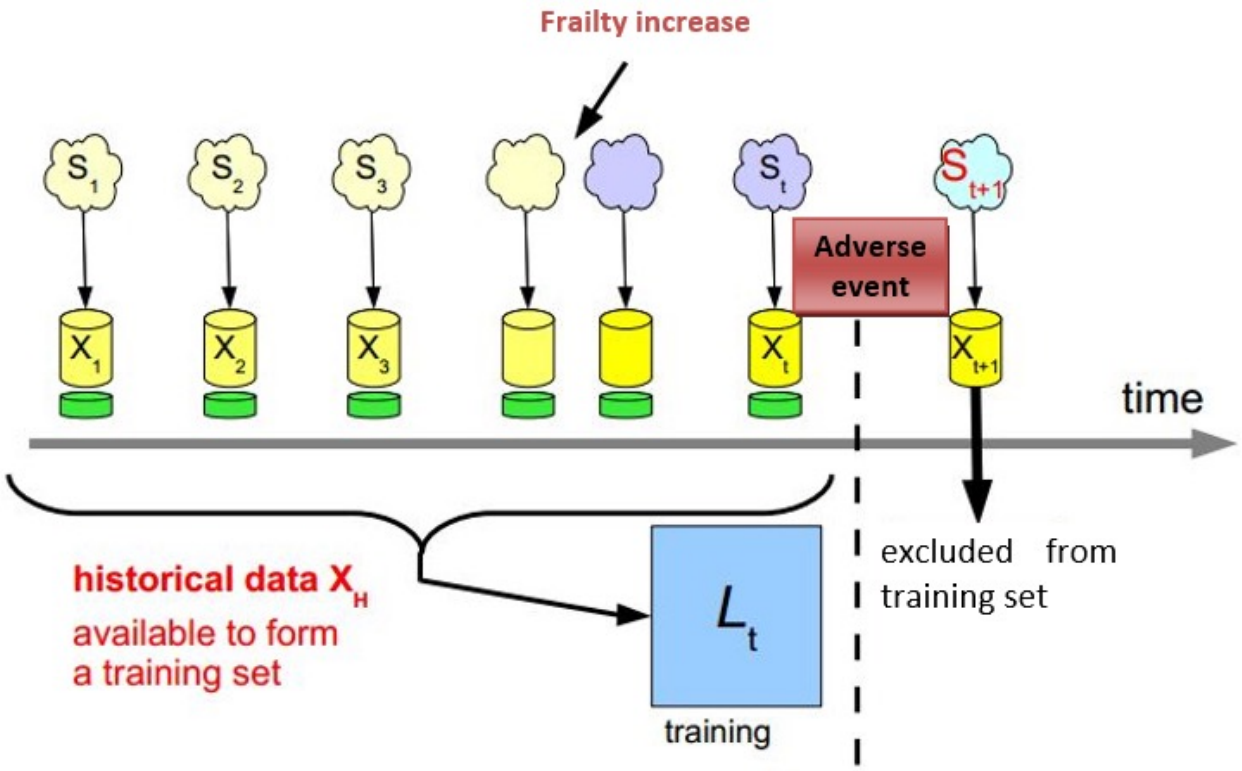
Decision support and alerts generation



Multi-parametric analysis towards prediction of frailty risk



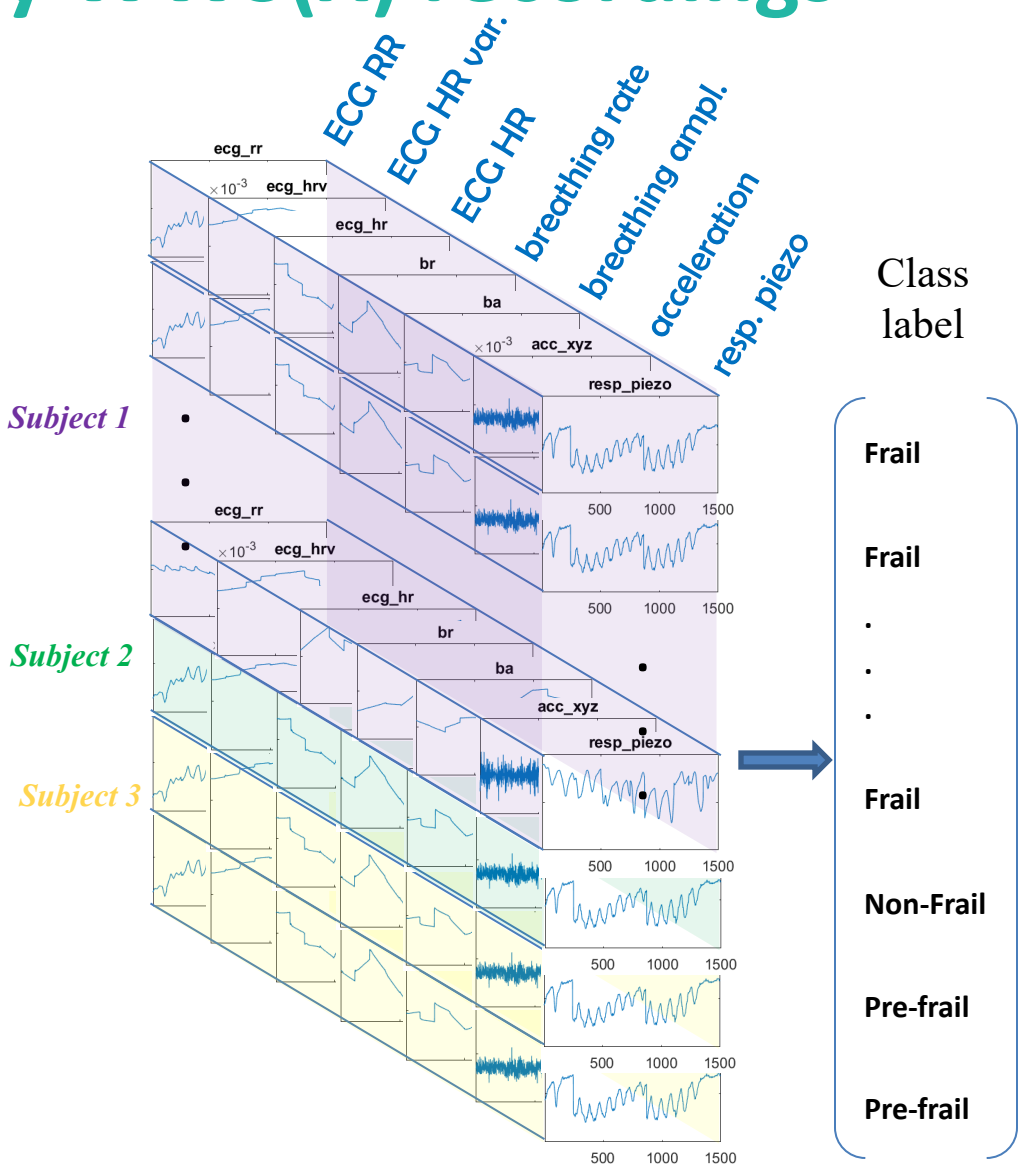
Prediction of adverse events



- If any instance indicates future event (positive class), the rest of the instances of the same subject are also considered as positive class
- Multiple Instance Learning (MIL) problem in which the temporal alignment of the multiple instances (sessions) was ignored
- SPEC_MIL: a specializing multi-instance learner that follows a generalization of the miSVM (MiSVM applies SVM for MIL)

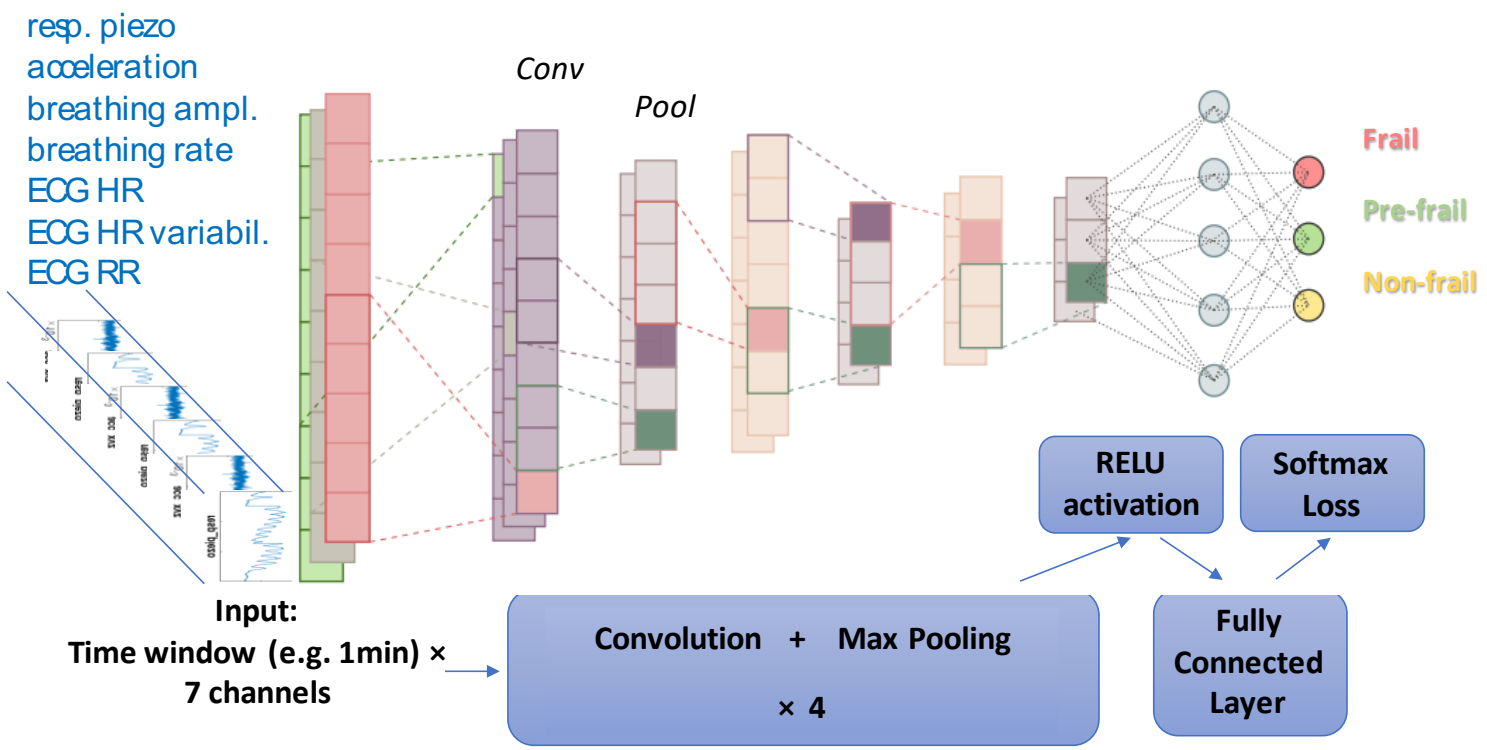
Predicting Fried by WWS(X) recordings

- **Data:** Concatenation of different signals to form a tensor for each subject and concatenation of tensors for all subjectss
- **Challenges:** Different # of frames across subjects and different # of subjects across classes
- **Method:** based on PARAFAC decomposition, multiple instance learning and Quadradic Discriminant Analysis classifier (TensMIL)
  - After data cleaning we represented the data using 3-D tensors.
  - Features were extracted by tensor decomposition techniques.
  - Frailty status prediction: Fusion of one class SVM models in MIL setting.



Complexity 2019

Predicting Fried by WWS(X) recordings



Deep convolutional neural networks (CNNs) developed for prediction of frailty status

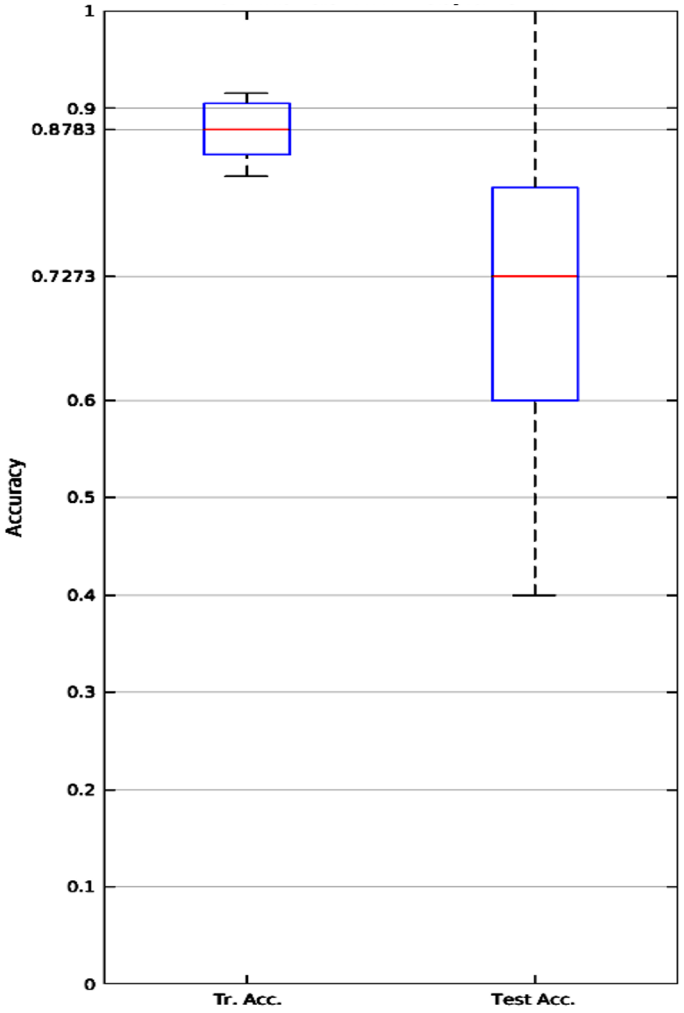
Complexity 2019

Predicting Fried by WWS(X) recordings

FrailSafe physiological measurements data (available until M24)

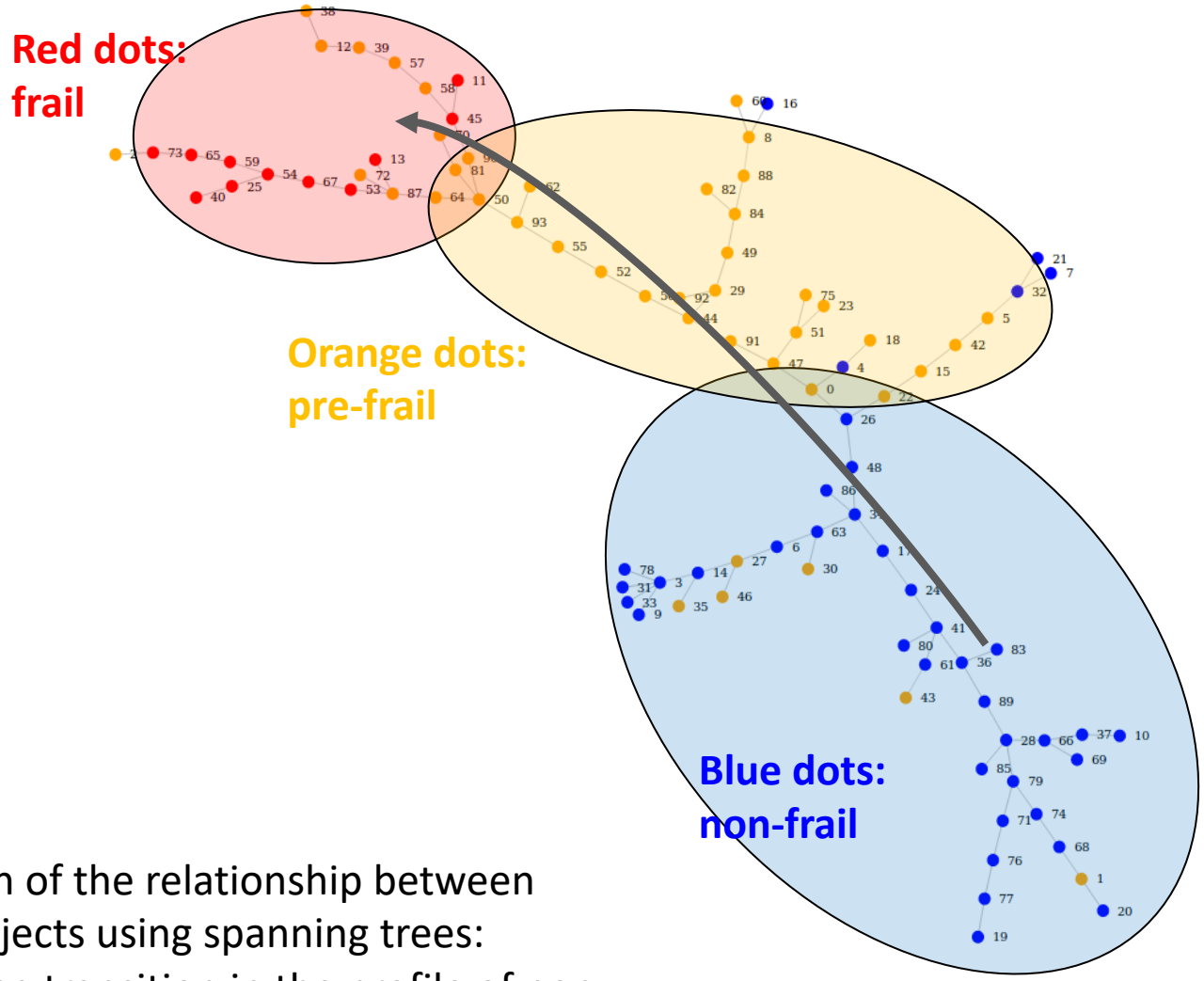
Subjects	**105**
Non-frail	44 (41.9%)
Pre-frail	49 (46.67%)
Frail	12 (11,43%)
Time windows (TW)	**10506**
TW (Non-frail)	3409 (32.45%)
TW (Pre-frail)	5216 (49.65%)
TW (Frail)	1881 (17.09%)

StrProxSGD, rank=60



Training and test median accuracy for 10-fold CV

Predicting Fried by WWS(X) recordings



Visualization of the relationship between training subjects using spanning trees: illustrates the transition in the profile of non-frail subjects to frail subjects.

Results: Prediction of adverse events

- Evaluation on 120 subjects
- Multiple Instance Learning
- Selection of results based on: AUC>0.6 and BAC>=0.64 for features combinations and compared always against clinical and Fried only

Raw features	AUC	Acc.	Balanced Acc. (BAC)
All (FS+clinical)	0.68	0.69	0.65
All (FS+clinical) no GPS or no Games	0.68-0.69	0.69-0.70	0.64-0.65
Clinical	0.60	0.70	0.63
Fried	0.65	0.70	0.57

Delta features	AUC	Acc.	Balanced Acc. (BAC)
All (FS+clinical), no GPS	**0.71**	**0.69**	**0.68**
WWSX+Games	**0.68**	**0.71**	**0.69**
Clinical	0.29	0.47	0.39
Fried	0.46	0.61	0.47

TENSMIL: tensor decomposition for MIL classification of multidimensional data

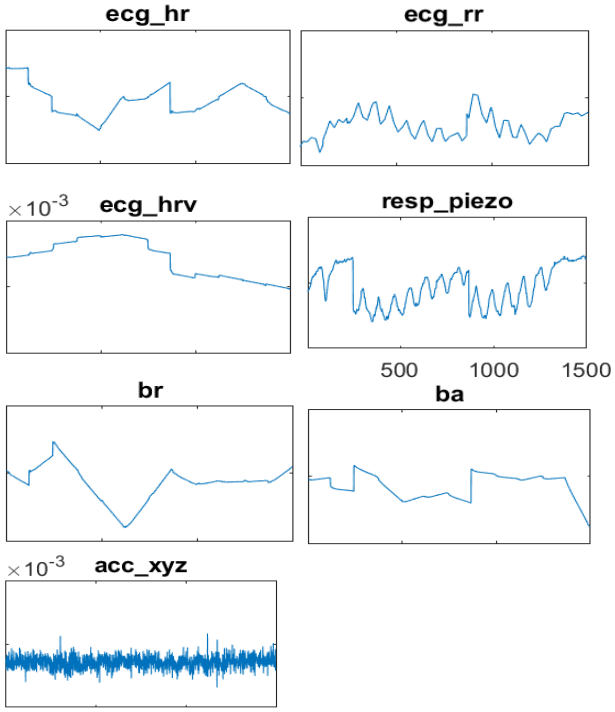
Papastergiou, T., E.I. Zacharaki, and V. Megalooikonomou, 2018, 2019.

MIL algorithm

TensMIL

Generalized feature extraction by tensor decomposition
StrProxSGD, ALS

TensMIL

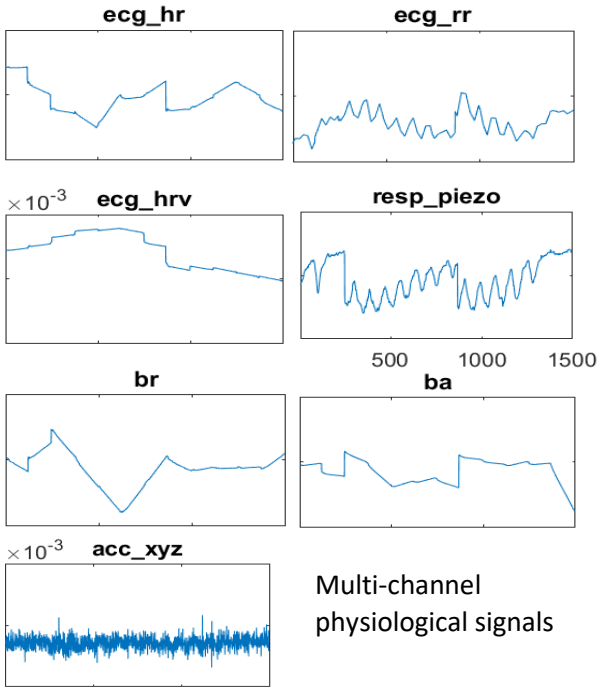


Frailty estimation

Efficient both on **fully or partially (10%) observed data** compared to state-of-the-art

Multiple Instance Learning (MIL)

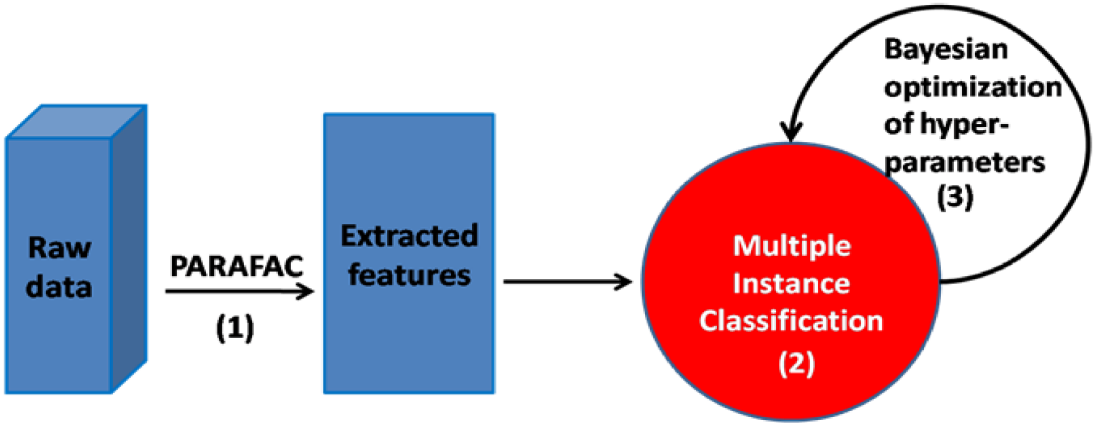
- **Classic ML**: each object represented by a feature vector
- **MIL**: each object represented by a collection of feature vectors
- We only know the annotation of the bags (objects)
- **The feature vectors of each object**: instances
- Class labels only for the objects
- MIL methods
  - In the space of instances
    - MI-SVM (Andrews et al. 2002)
  - In the space of objects
    - (Gärtner et al. 2002)
  - In nested space
    - MILES (Chen et al. 2006) , JC2MIL (Sikka et al. 2015)



Multi-channel physiological signals

TensMIL[1]: Method overview

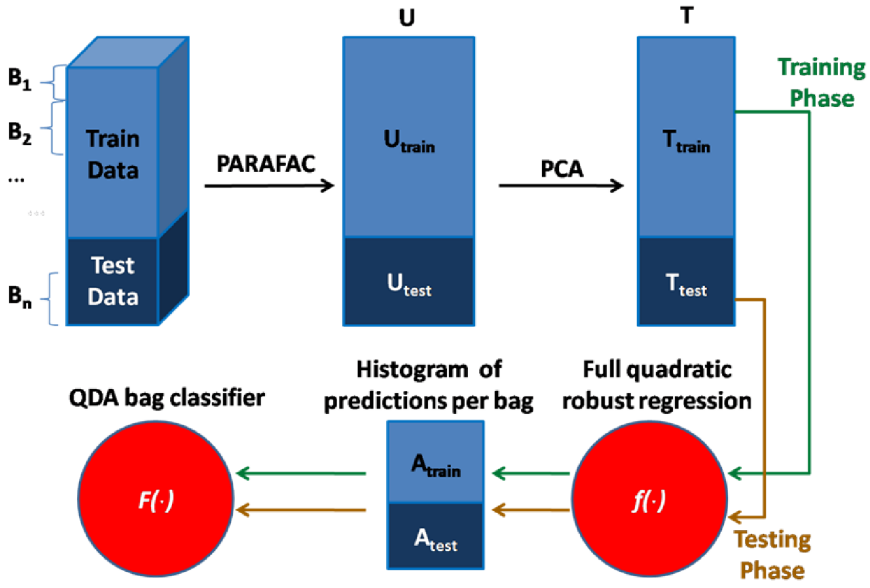
1. **Data representation and feature extraction**
   - Computation of a multidimensional dictionary with the CANDECOMP/PARAFAC decomposition
2. **MIL**
   - Sequential discrete models n the space of instances and in the space of objects
3. **Optimization**
   - For the learning of hyper-parameters.



[1](Papastergiou et al. 2018)

TensMIL

1. Feature extraction
2. Training using only instances (with inaccurate labels – we know only the labels of the bags (whole bag, i.e., person)
3. Fusion of the outcome using probabilistic modelling (histograms)
4. Classification of the objects

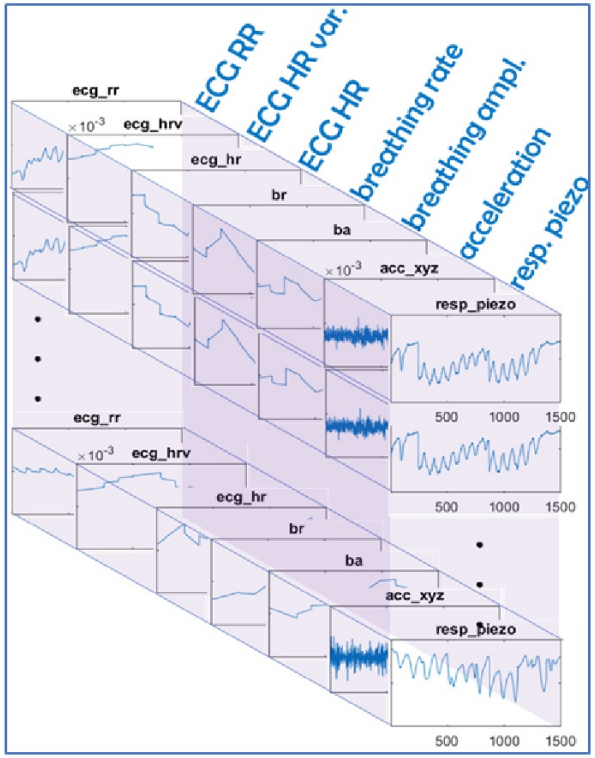


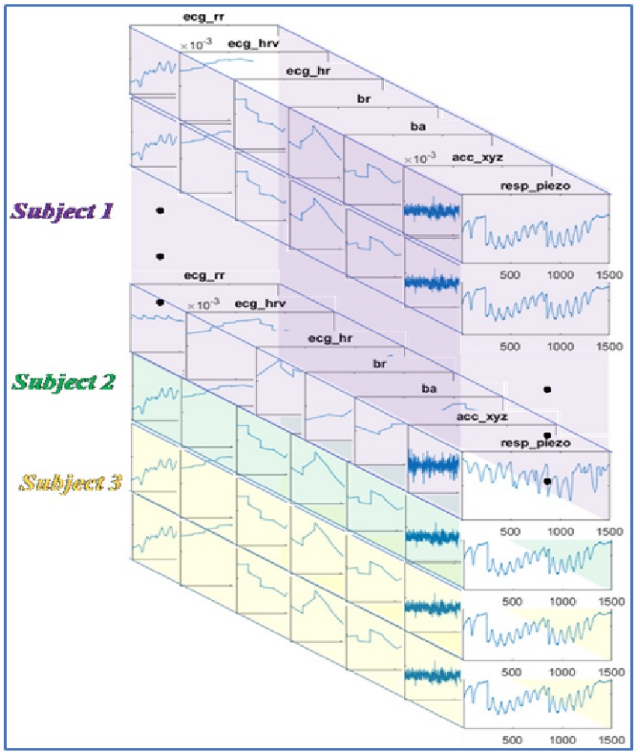
Architecture of TensMIL

U: feature matrix extracted from raw data by PARAFAC decomposition
T: score matrix obtained by performing PCA on U
A: matrix containing the bag-level features
f(·): full quadratic regression model
F(·) Quadratic Discriminant Analysis (QDA) classifier.

Experiments: Dataset

- Physiology signals from monitoring of older people (FrailSafe)
  - Non-overlapping windows of 1 minute (1500 time points)
  - **Tensor**: $19244 \times 1500 \times 7$





Class	Number of objects	Number of instances	Percentage of objects	Percentage of instances
Non-frail	49	7127	42.24%	37.03%
Pre-frail	54	8803	46.55%	45.74%
Frail	13	3314	11.21%	17.22%
Total	116	19244	100%	100%

Results

Μέθοδος	ALS R=60		StrProxSGD R=60 (90% ελλιπείς τιμές)	
	Acc	Bacc	Acc	Bacc
TensMIL	45.76(0.13)	34.06(0.09)	73.41(0.01)	67.17(0.13)

Μέθοδοι	Χρόνος εκπαίδευσης	Χρόνος ελέγχου
MILES	42 sec	1 sec
JC2MIL	56 sec	<1sec
MILBoost	52 sec	5 sec
MCILBoost	309 sec	6 sec
TensMIL	6 sec	<1 sec

Μέθοδος	ALS R=60	StrProxSGD R=60 (90% ελλιπείς τιμές)
MILES	51.59(0.13)	67.20(0.11)
JC2MIL	56.82(0.07)	55.30(0.08)
MILBoost	50.83(0.15)	54.39(0.15)
MCILBoost	45.46(0.14)	60.91(0.22)
TensMIL	**54.02(0.13)**	**80.83(0.16)**

- Three class classification problem
  - With partially observable data the balanced accuracy is 33.11% more efficient
- Two class classification problem
  - With partially observable data the balanced accuracy is 13.63% - 26.44% more efficient
- Training and testing time
  - **Feature extraction** : ~2.5 hours
  - **Training time TensMIL**: 7 – 72 times faster
  - **Test time TensMIL**: < 1 sec

Frailsafe Consortium

Smartex, S.R.L.
ITALY

BrainStorm Multimedia
SPAIN

AGE Platform Europe
BELGIUM

CERTH/ITI
GREECE

AgeCare Ltd
CYPRUS

University of Patras
GREECE

Gruppo SIGLA S.R.L.
ITALY

HYPERTECH S.A.
GREECE

University Hospital (CHU) of Nancy and INSERM
FRANCE

рахмат
danke
謝謝
ngiyabonga
teşekkür ederim
Баярлалаа
mersi
kia ora
barka
welalin
tack
spas
спасибо
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
mahalo
tapadh leat
thank you
gracias
хвала
asante
manana
obrigada
tenki
kiitos
dankie
nanni
nandri
dhanyavad
mauruuru
köszönöm
bedankt
bayarlalaa
gràçie
hvala
enkosi
akun
dankon
ačiū
djiere dieuf
tau
mochchakkeram
murakoze
mamnun
дякую
chokrane
dziękuję
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
go raibh maith agat
sobodi
dėkuji
obrigado
sukriya
kop khun krap
sagolun
arigatô
takk
dakujem
trugarez
mèsi
didi madloba
najis tuke
ありがとう
tanemirt
grazie
rahmet
kam sah hamnida
rahmat
terima kasih
ευχαριστώ
diolch
dhanyavadagalu
shukriya
merće
мерси
তোমাকে ধন্যবাদ
감사합니다
xièxie
merci