Novelty Detection in Data Streams

Rita P. Ribeiro   João Gama

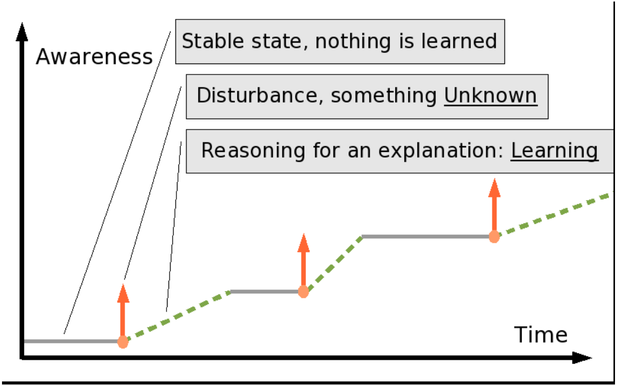
LIAAD-INESC TEC, University of Porto, Portugal

Outline

Outline

Outline

How do we learn?



Novelty

- **Novelty** is a relative concept defined in the context of a representation of our current knowledge
- **Novelty Detection** refers to the automatic identification of unforeseen phenomena embed in a large amount of normal data
- Specially useful when novel concepts represent abnormal or unexpected conditions
  - expensive to obtain abnormal examples
  - probably impossible to simulate all possible abnormal conditions

Outline

Applications

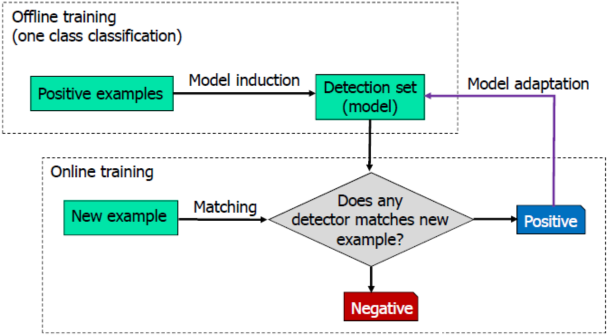
- Intrusion Detection
- Fault Detection
- Fraud Detection
- Medical Diagnosis
- Spam Filter
- Text Classification

Outline

Outline

One-Class Classification: **Problem Definition**

- Offline Phase
  - Normal concept is composed by one class.
  - All training examples belong to the normal class.
- Online Phase
  - Examples not explained by the normal concept are labeled as abnormal.



Outline

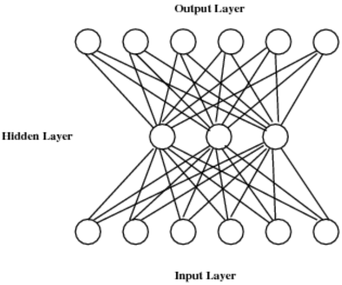
One-Class Classification: **Common Approaches**

- Some techniques use:
  - Artificial Neural Networks
  - Support Vector Machines
  - kNN based approaches
  - Kernel based approaches
  - Parzen windows

One-Class Classification: **Common Approaches - II**

Autoencoders [Japkowicz, 1999]

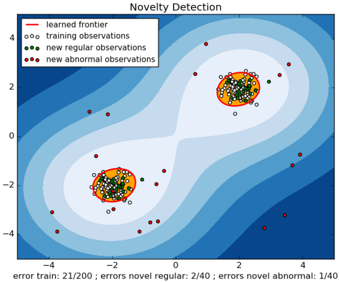
- three layer network
- nr. of neurons in the output layer is equal to the input layer
- the network is trained with backpropagation to reproduce the input at the output layer
- difference between the input example and the output:
  - $<$ *threshold*: example is from normal class
  - otherwise: is a counter-example of normal class



One-Class Classification: **Common Approaches - III**

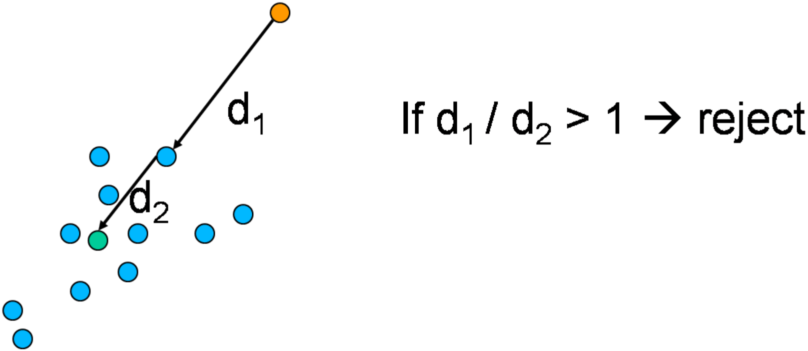
Support Vector Data Description [Tax and Duin, 2004]

- obtains a spherical boundary, in the feature space, around the data
- the volume of this hypersphere is minimized, to reduce the effect of incorporating outliers in the solution
- examples lying outside the hypershere are considered abnormal



One-Class Classification: Nearest neighbor

Nearest neighbor for novelty detection (Tax, 2001)



Outline

One-Class Classification: Case Study: Predict Train Door Failures

Sequential anomalies: a study in the Railway Industry [Ribeiro et al., 2016]

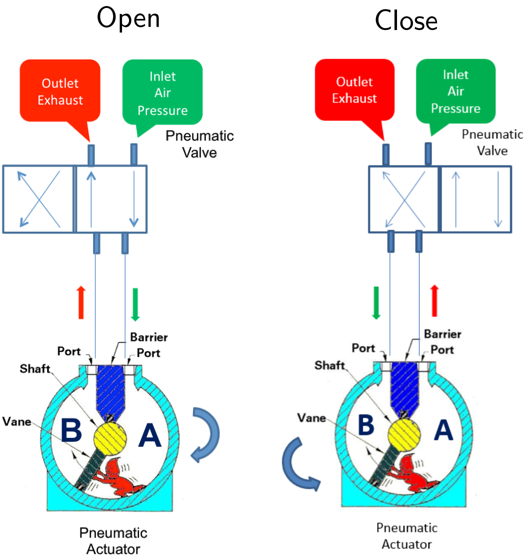
- Create a system to anticipate the development of a door failure to proper maintenance schedule and to avoid breakdown.
- Find patterns in data that do not correspond to the expected behaviour.

- The goal is **NOT** to detect an **abnormal door movement**.
- Issue an alarm whenever a **structural door failure** is about to happen.
- **Detect sequential anomalies**.

One-Class Classification: Case Study: Predict Train Door Failures - II

Original Data

- Pressure readers at both chambers.
- Interval: 1/10 sec.
- Episodes time-series data set.
- $\approx$ 60 observations / episode.
- Sept. to Dec. 2012.
- $\approx$ 500.000 observations.
- Movement duration not steady.
- Unlabelled data set.



One-Class Classification: **Case Study: Predict Train Door Failures - III**

- Problem Setting
  - Data: sequence of cycles (Open or Close).
  - Goal: predict the structural state of the door: Normal or Failure.
- Two-Step Approach:
  1. **Abnormal Cycle Detection**
     - classify each cycle as Normal (1) or Abnormal (0)
  2. **Failure Sequence Detection**
     - classify sequence of cycles as Normal or Failure.

One-Class Classification: **Case Study: Predict Train Door Failures - IV**

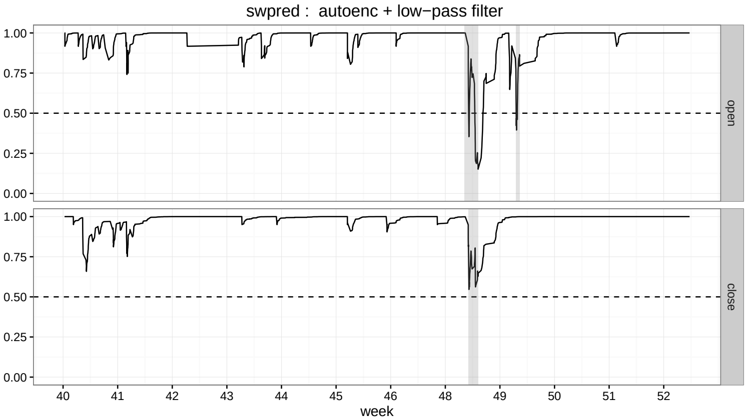
- Abnormal cycle detection assumes independence between observations.
- Structural failure detection must take into account sequence of observations.
- For that, we use a **Low-Pass Filter** to post-process cycle classification output.

$$y_i = \begin{cases} 1 & \text{if } i = 0 \\ y_{i-1} + \alpha * (x_i - y_{i-1}) & \text{if } i > 0 \end{cases}$$

  where, for instant i, y_i is filter output and x_i is original signal.
- The α parameter smoothes abrupt changes in the original signal.
- Lower values of α cause more inertia.
- Failure Threshold: $y_i < 0.5$

One-Class Classification: **Case Study: Predict Train Door Failures - V**

- Experiments
  - different memory models
  - different one-class classification models



One-Class Classification: Case Study: Predict Train Door Failures - VI

Impact of low-pass filter

- reduction of false alarms using a sliding window with self-feedback memory model.

	Number of False Alarms					
	open			close		
learning algorithm	before filter	after filter	reduction	before filter	after filter	reduction
autoenc	34	6	82%	30	0	100%
boxplotEns	146	36	75%	115	10	91%
ocsvm	292	99	66%	325	216	34%
occ	220	39	82%	239	16	93%

- low-pass filter significantly reduces the number of false alarms.

Outline

Outline

Novelty Detection: **Problem Definition**

- Training set (Offline Phase)
  - $D_{tr} = (X_1, y_1), (X_2, y_2), \ldots, (X_m, y_m)$
  - X_i: vector of input attributes for the ith example
    y_i: target attribute
  - $y_i \in Y_{tr}$ where $Y_{tr} = c_1, c_2, \ldots, c_L$
- When new data arrive (Online Phase)
  - Given a sequence of unlabelled examples X_{new}
  - Goal: Classify X_{new} in Y_{all} where $Y_{all} = c_1, c_2, \ldots, c_L, \ldots, c_K$ and $K > L$

Open-set Recognition

Anuran species recognition using a hierarchical classification approach

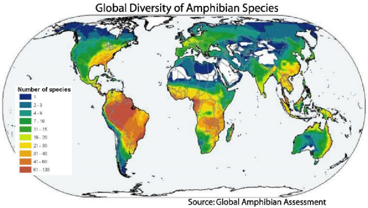
Juan G. Colonna[1,2], João Gama[2], and Eduardo F. Nakamura[1]

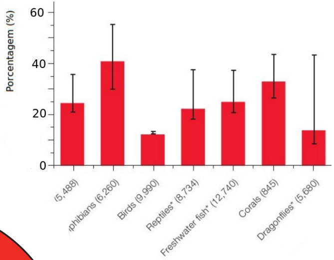
[1]Federal University of Amazonas (UFAM), Institute of Computing (Icomp)
[2]Laboratory of Artificial Intelligence and Decision Support (LIAAD), INESC Tec
{juancolonna, nakamura}@icomp.ufam.edu.br
jgama@fep.up.pt

Getting more from family, genus and species of frogs

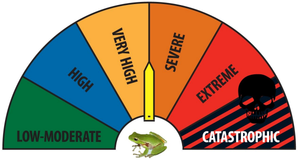
Introduction - Why frogs?

- **Anura** is the name of an order of animals in the **Amphibian** class which lack a tail, this includes **frogs** and **toads**.





- **Frogs are very sensitive to environmental changes**



2

Why monitor populations of frogs?

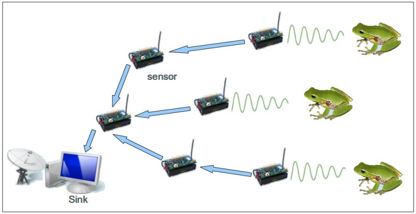
Hypothesis: Tracking the changes in the anuran populations can help us to determine ecological problems in early stages.

It involves several manual tasks!

3

Proposal

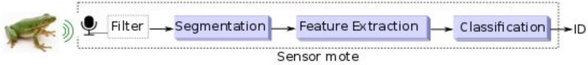
Signal processing (SP) + Wireless Sensor Networks (WSN) + Machine Learning (ML)



Advantages: It is Automatic, less intrusive and allows long term monitoring.

4

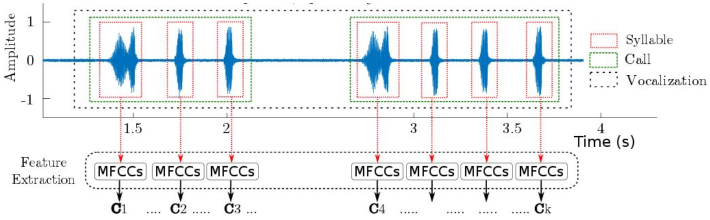
How to do that?

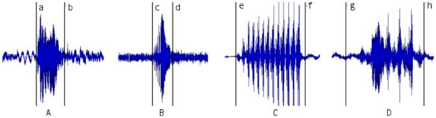


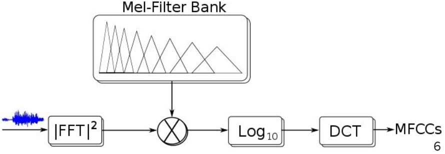
1) Pre-processing:
   a) Filter: band-pass filter, wavelet decomposition, etc.
   b) Segmentation: **syllable-based approach** (x_k)
2) Feature Extraction: that maps $x_k \rightarrow c_k$
   a) Mel-frequency cepstral coefficients (MFCCs)
   b) Spectral centroid, Spectral bandwidth, Pitch, etc.
3) Recognition: ML technique to classify $c_k \rightarrow$ID (species ID)
   a) Support Vector Machine, kNN, Tree, etc.

5

Segmentation and feature extraction

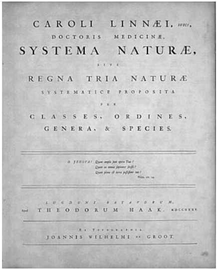






Knowledge organization

Carl Linnaeus has defined a particular form of biological organization called *taxonomy* in his work *Systema Naturae* (1735).



CAROLI LINNÆI,

DOCTORIS MEDICINÆ,

SYSTEMA NATURÆ,

REGNA TRIA NATURÆ

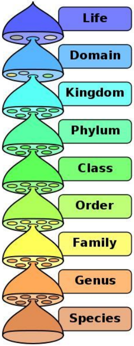
SYSTEMATICE PROPOSITA

CLASSES, ORDINES,

GENERA, & SPECIES.

THEODORUM HAAK.

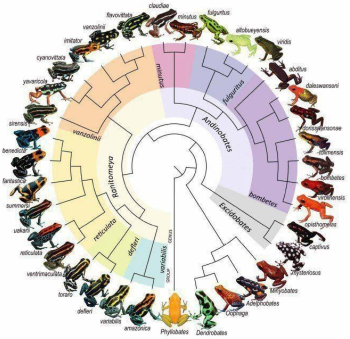
JOANNIS WILHELMI GROOT.



How to improve the classification using the taxonomy?

- The anura Order has 31 Families (approximately)
- These Families are divided into several genus
- And finally, these genus are divided in almost 6000 species

Hypothesis: the phylogenetic taxonomy may describe similar calls among species that belong to the same genus and family[2].



Illustrative figure.

[2] B. Gingras and W. T. Fitch.
A three-parameter model for classifying anurans into four genera based on advertisement calls.
The Journal of the Acoustical Society of America, 133(1):547–559, 2013.

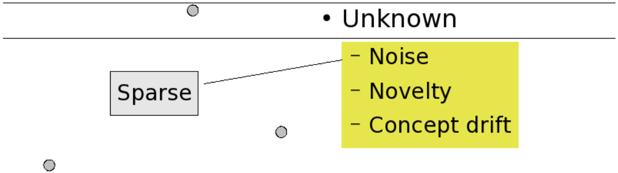
Novelty Detection: **Problem Definition**

- Offline Phase
  - All training examples belong to the *known* classes.
- Online Phase
  - Examples not explained by the current model are labeled as *unknown*.
  - Cohesive group of *unknown* examples are used to detect *novel* classes or extensions to the *known* classes.

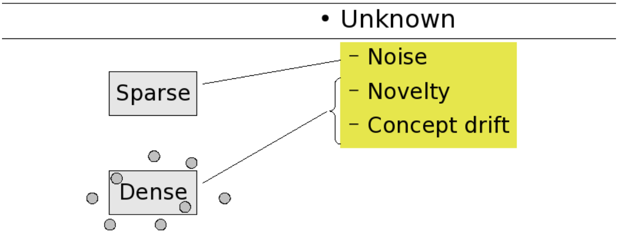
Novelty Detection: **Problem Definition - II**

- In data streams, concepts are hardly ever constant.
- It is important **to distinguish novelty from**:
  - noise and outliers
  - concept drift
  - concept evolution
  - recurring concepts

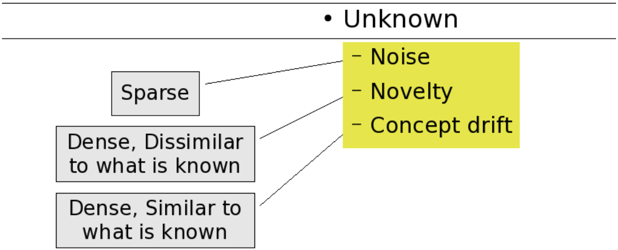
Novelty Detection: **Problem Definition - III**



Novelty Detection: **Problem Definition - III**



Novelty Detection: **Problem Definition - III**



Novelty Detection: **Problem Definition** - IV

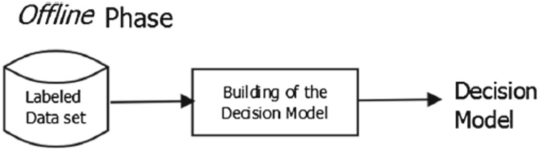
- In data streams scenarios:
  - new concepts may appear
  - known concepts may evolve, disappear or reappear
- By monitoring the data stream, emerging concepts may be discovered
- Emerging concepts may represent
  - an extension to a known concept (extension)
  - a novel concept (novelty)

Novelty Detection: **Problem Definition** - VI

- Novelty Detection Systems
  - **OLINDDA**: OnLIne Novelty and Drift Detection Algorithm [Spinosa et al., 2007]
  - **ECSMiner**: Enhanced Classifier for data Streams with novel class Miner [Masud et al., 2011]
  - **MINAS**: MultI-class learNing Algorithm for data Streams [de Faria et al., 2016]

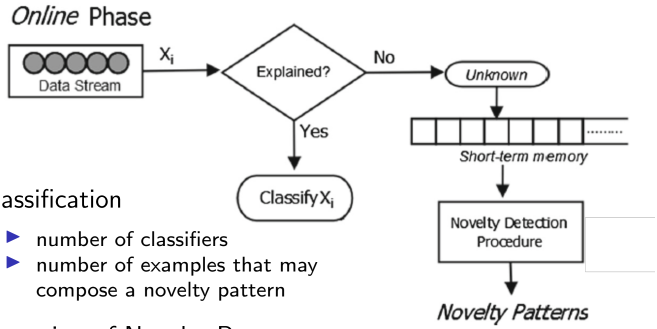
Outline

Novelty Detection: **Key Aspects**



- Decision Model for Known Normal Patterns
  - number of classes that represent the normal patterns
  - number of classifiers
  - supervised/unsupervised learning

Novelty Detection: **Key Aspects - II**

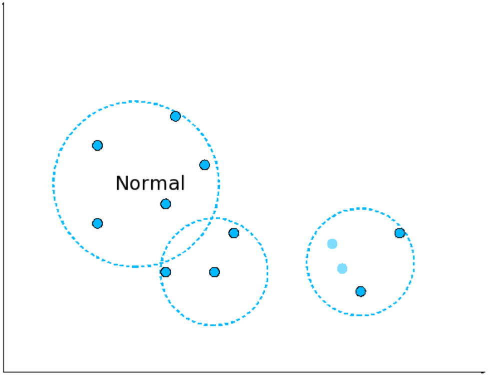


- Classification
  - number of classifiers
  - number of examples that may compose a novelty pattern
- Detection of Novelty Patterns
  - number of classes that compose the novelty patterns
- Decision Model Update
  - type of update (with/without feedback)
  - number of classifiers
  - forgetting mechanisms to remove outdated concepts

Outline

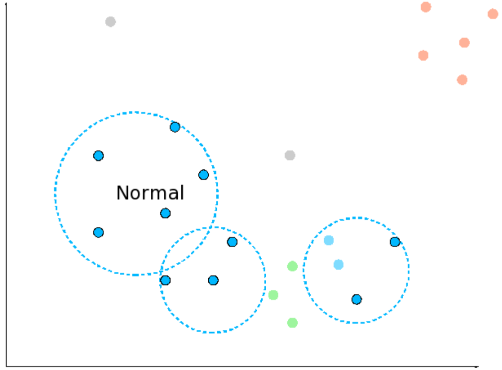
Novelty Detection: **MINAS algorithm**

MINAS: MultI-class learNing Algorithm for data Streams [de Faria et al., 2016]



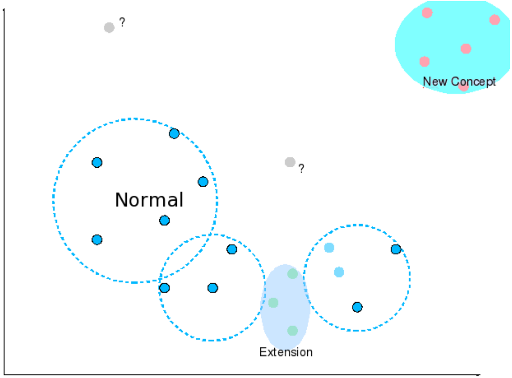
Novelty Detection: **MINAS algorithm**

MINAS: MultI-class learNing Algorithm for data Streams
[de Faria et al., 2016]



Novelty Detection: **MINAS algorithm**

MINAS: MultI-class learNing Algorithm for data Streams [de Faria et al., 2016]



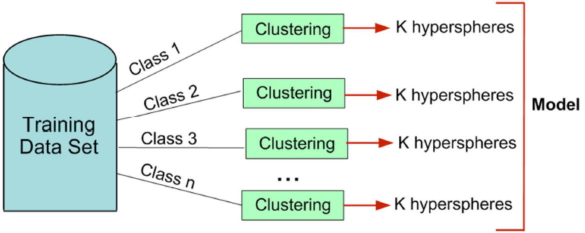
Novelty Detection: **MINAS algorithm - II**

- unsupervised algorithm for novelty detection in data streams multi-class problems
  - training examples are composed by many classes
  - there may be also several novel classes
- use of offline (training) and online phases
  - in each phase learns one or more classes
- cohesive set of examples is necessary to learn new concepts or extensions
  - isolated examples are not considered as novelty

Novelty Detection: **MINAS algorithm**- III

Offline Phase

- learns a decision model based on the known concepts about the problem (k-means or Clustream)
- runs only once
- each class is represented by a set of clusters (hyperspheres)



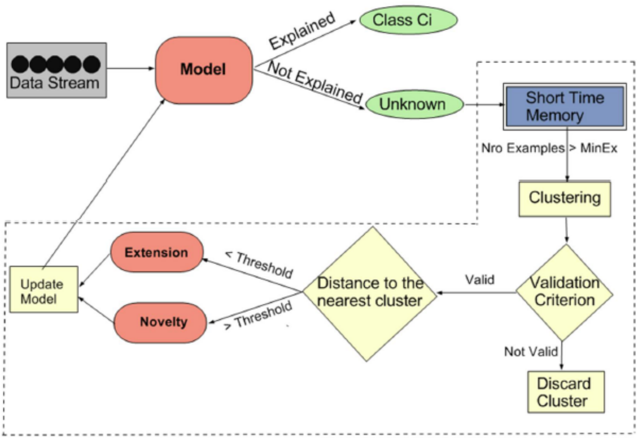
Novelty Detection: **MINAS algorithm**- IV

Online Phase

- receives new unlabelled examples from the stream
- classifies each new example as one of known classes or as unknown
- unknown examples are stored in the Short Term Memory
- from time to time
  - finds clusters in the examples stored in the Short Term Memory
  - clusters far away from existing ones: novel concept.
  - clusters close to existing ones: extend known concepts.

Novelty Detection: **MINAS algorithm** - V

Online Phase



Novelty Detection: **MINAS algorithm** - VI

Treatment of Outliers

- clustering is applied to the unknown examples
- each cluster is validated by the evaluation of its representativeness and cohesiveness
- clusters with low value are considered invalid and removed
- however, their examples stay in a temporary memory
- if there is no space available, the oldest example is removed
- there is a high chance that the removed examples are noise or outliers

Outline

Evaluation Issues: **Adaptation of binary classification metrics**

- *Precision* and *Recall* [Albertini and de Mello, 2007]

$$Precision = \frac{\#\ true\ detected\ novelties}{\#\ detected\ novelties} \qquad Recall = \frac{\#\ true\ detected\ novelties}{\#\ novelties}$$

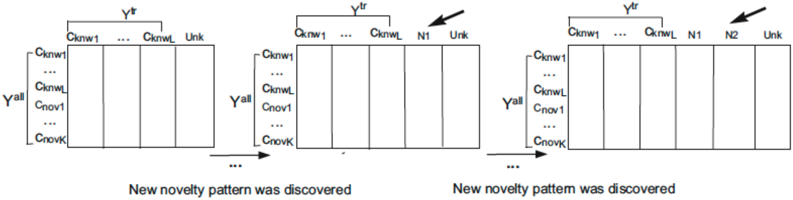
- *Mnew* and *Fnew* [Spinosa et al., 2007]

$$Mnew = \frac{100 * \#\ false\ detected\ novelties}{\#\ novelties} \qquad Fnew = \frac{100 * \#\ false\ detected\ normalities}{\#\ total\ -\ \#\ novelties}$$

- How to reflect the *unknown* label in the evaluation?
- How to extend the evaluation to a multi-class scenario?

Evaluation Issues: **A Rectangular Confusion Matrix** [de Faria et al., 2016]

- At the beginning of online phase, the model is composed by the classes learned offline.
- When a new concept is discovered, the model is updated and a new column is added to the confusion matrix.



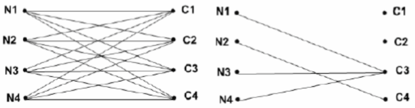
- rows: true classes (known + novelty)
- columns: predicted classes (known + novelty patterns + unknown)

Evaluation Issues: **A Rectangular Confusion Matrix** [de Faria et al., 2016]

- The confusion matrix is not squared.
- The number of columns increases dynamically
- From time to time novelty patterns are associated to classes so to minimize the error.

	C_1	C_2	N1	N2	N3	N4	Desconhecido
C_1	6000	2900	0	800	50	0	100
C_2	2000	5200	700	0	30	0	50
C_3	300	450	1950	100	100	50	50
C_4	200	300	1900	1400	100	20	100

Matriz de Confusão



Grafo bipartido correspondente

Subgrafo bipartido resultante

Outline

Challenges and Future Work

- Most of the techniques use one-class classification
  - many real-world applications are, in fact, multi-class scenarios
- Assumption that true labels will be available
  - time consuming
  - support from domain expert
- Outliers, noise, changing environments
  - depends on the data set
  - concepts may evolve gradually or abruptly
  - distinguish noise, outliers from a novel concept

Challenges and Future Work - II

- Recurring contexts
  - an important phenomenon observed in many real-world applications (e.g. climate change, electricity demand)
  - systems typically use a forgetting mechanism of old concepts;
  - a recurring class may be confused with the emergence of a new class $\rightarrow$ it leads to high false positive rates
  - relearn an old concept is a waste of effort
  - ideally, they should be saved and reexamined at some later time
  - identify when a concept is reappearing

Challenges and Future Work - III

- When to apply novelty detection in data streams
  - whenever new example arrives is time consuming
  - define the time interval
- Algorithms to induce the decision model
  - supervised algorithms need labeled examples
  - unsupervised algorithms (e.g. kmeans) assume that classes constitute hyperspheres, need nr. clusters as input., handle only numerical attributes
- Evaluation issues and experimental methodology
  - lack of standards

Some References I

Abdallah, Z. S., Gaber, M. M., Srinivasan, B., and Krishnaswamy, S. (2016).
Anynovel: detection of novel concepts in evolving data streams.
Evolving Systems, 7(2):73–93.

Albertini, M. K. and de Mello, R. F. (2007).
A self-organizing neural network for detecting novelties.
In *Proceedings of the 2007 ACM Symposium on Applied Computing*, SAC '07, pages 462–466, New York, NY, USA. ACM.

Costa, B. S. J., Angelov, P. P., and Guedes, L. A. (2014).
Real-time fault detection using recursive density estimation.
Journal of Control, Automation and Electrical Systems, 25(4):428–437.

de Faria, E. e., Ponce de Leon Ferreira Carvalho, A., and Gama, J. a. (2016).
MINAS: multiclass learning algorithm for novelty detection in data streams.
Data Mining and Knowledge Discovery, 30(3):640–680.

Faria, E. R., Gonçalves, I. J. C. R., de Carvalho, A. C. P. L. F., and Gama, J. (2016).
Novelty detection in data streams.
Artificial Intelligence Review, 45(2):235–269.

Japkowicz, N. (1999).
Concept-learning in the absence of counter-examples: An autoassociation-based approach to classification.

Krawczyk, B. and Woźniak, M. (2015).
One-class classifiers with incremental learning and forgetting for data streams with concept drift.
Soft Computing, 19(12):3387–3400.

Masud, M. M., Gao, J., Khan, L., Han, J., and Thuraisingham, B. M. (2011).
Classification and novel class detection in concept-drifting data streams under time constraints.
IEEE Trans. Knowl. Data Eng., 23(6):859–874.

Some References II

Pereira, P., Ribeiro, R. P., and Gama, J. (2014).
Failure prediction - an application in the railway industry.
In *Discovery Science - 17th International Conference, DS 2014*, pages 264–275.

Pimentel, M. A. F., Clifton, D. A., Clifton, L., and Tarassenko, L. (2014).
A review of novelty detection.
Signal Process., 99:215–249.

Ribeiro, R. P., Pereira, P., and Gama, J. (2016).
Sequential anomalies: a study in the railway industry.
Machine Learning, pages 1–27.

Spinosa, E. J., de Leon Ferreira de Carvalho, A. C. P., and Gama, J. (2007).
Olindda: a cluster-based approach for detecting novelty and concept drift in data streams.
In Cho, Y., Wainwright, R. L., Haddad, H., Shin, S. Y., and Koo, Y. W., editors, *SAC*, pages 448–452. ACM.

Tax, D. M. J. and Duin, R. P. W. (2004).
Support vector data description.
Machine Learning, 54(1):45–66.