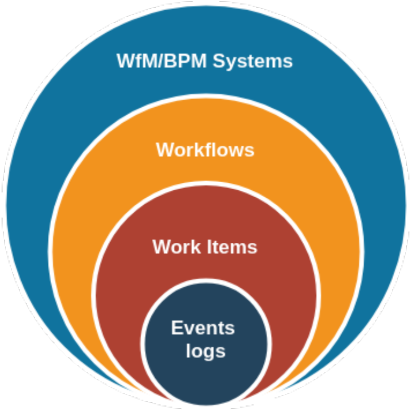
Automatic classification of event logs sequences for failure detection in WfM/BPM systems

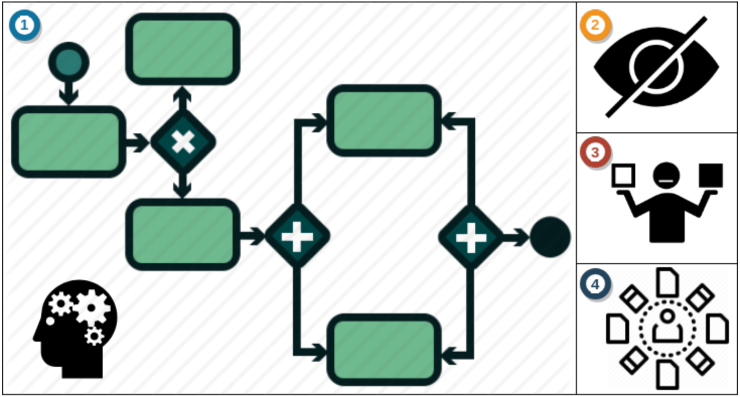
Johnnatan Jaramillo and Julián Arias-Londoño
Universidad de Antioquia

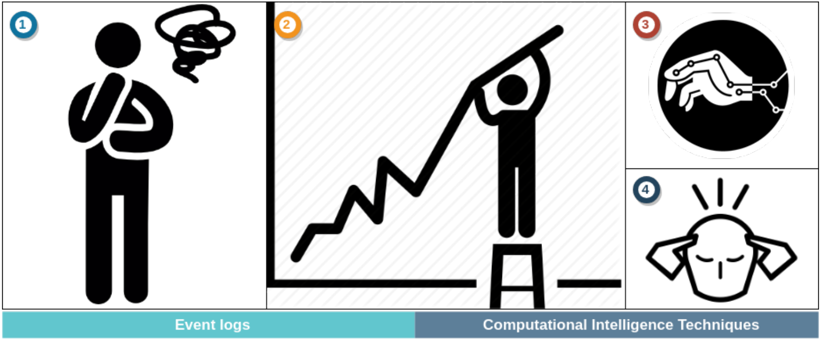
Outline

Business Process Management and Problem Context

It is a **methodology** that seeks to control, analyze, and improve organizational processes.




1
2
3
4


1
2
3
4
Event logs
Computational Intelligence Techniques

Problem:

Identify which work items are there in an error or failure state, given the large number of active work items that could be active in the system.

Problem:

Identify which work items are there in an error or failure state, given the large number of active work items that could be active in the system.

Proposed Solution:
To use techniques of Computational Intelligence and Machine Learning to detect failures in a business process, and to predict which of these work items will end up in an error state. All this from event logs.

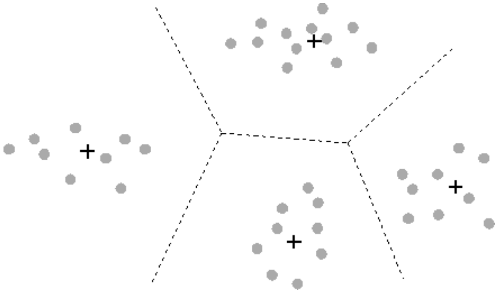
Characterization of event logs

- e.g. *Operation identifier*, *Event type*, *Originator Identifier*.
- **One-Hot Encoding**.

Workflow Id	Event Type				Operation Id				Integer
	1	2	3	4	1	2	3	4	
208aefee-e047	1	0	0	0	0	0	1	0	130
208aefee-e047	0	1	0	0	1	0	0	0	72
208aefee-e047	0	0	1	0	0	1	0	0	36
208aefee-e047	0	0	0	1	0	0	0	1	17

Required to:

- **Hidden Markov Model.**
- **Hidden *semi*-Markov Model.**
- **Non-stationary Hidden *semi*-Markov Model.**



Modeling

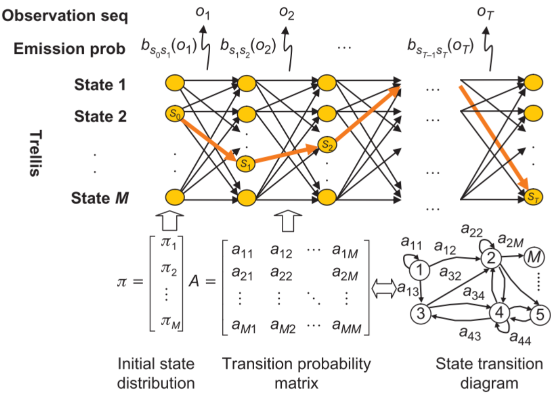


Figure: A standard Hidden Markov Model, taken from [2].

Hidden *semi*-Markov Model (HSMM)

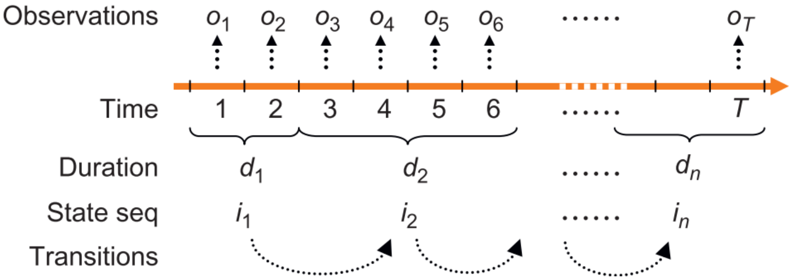


Figure: A general Hidden *semi*-Markov Model, taken from [2].

- Transition probability matrix ***"A"***.
- Emission observation probability matrix ***"B"***.
- Initial probabilities vector *"π"*.
- Set of hidden states with size ***"M"***.
- Observation dictionary with size ***"k"***.

HMM:

- $a_{ij} \equiv P\left[S_t = j | S_{t-1} = i\right]$

Comparison (2)

- Discrete random variable ***"D"***.

HSMM:

- $a_{(i)(j,d)} \equiv P\left[S_{[t+1:t+d]} = j | S_{t]} = i\right]$

NHSMM:

- $a_{(i,j)(d)} \equiv P\left[S_{t+1} = j | S_{[t-d+1:t} = i\right]$

Performance

- An important consideration when using ML methods is his **computational complexity**.
- In a HMM, the complexity in terms of memory is $O(MT)$.
- And the complexity in terms of time is $O(M^2T)$.
- In HSMM, there is a computational complexity of $O((M^2 + MD^2)T)$ during training.
- In NHSMM is of $O(M^2TD^2)$ during training.
- The implementations of HMM and HSMM models used in this work were made in **Apache Spark**.
- With this implementation the performance of the algorithms can be **scalable**.

Experiments and Results

- The event logs were generated by a **WfM system**.
- This system supports a **real-life** business process related to banking.
- The dataset contains around **sixty millions** of event logs, that correspond to **460,000** event logs sequences.
- There are **two groups** of sequences: a group that finish successfully, and a group that finish incorrectly o with some error.
- **60%** of sequences correspond to the first group, and the remaining **40%** to the second group.

- **Cross-validation** was performed with five folds.
- The parameter $\boldsymbol{M}$ was varied between *10* and *50*, and the parameter $\boldsymbol{D}$ between *3* and *6*.
- The classification performance indicators are calculated as follows:
  - ***sensitivity:*** $tp/(tp + fn)$
  - ***specificity:*** $tn/(tn + fp)$
  - ***accuracy:*** $(tp + tn)/\#observations$
  - ***geometric mean:*** $\sqrt{sensitivity * specificity}$

Results HMM

k-means and $k = 57$

M	*Sensitivity*	*Specificity*	*Accuracy*	*G-mean*
40	45.33%	74.46%	62.52%	**58.10%**

k-mode and $k = 300$

M	*Sensitivity*	*Specificity*	*Accuracy*	*G-mean*
40	45.31%	74.46%	62.51%	**58.08%**

Without clustering and $k = 4039$

M	*Sensitivity*	*Specificity*	*Accuracy*	*G-mean*
20	74.35%	70.37%	72.00%	**72.34%**

Results HSMM without clustering, $M = 10$ and $k = 4039$

D	*Sensitivity*	*Specificity*	*Accuracy*	*G-mean*
3	78.98%	72.97%	75.43%	75.70%
4	**99.90%**	**77.49%**	**86.70%**	**87.72%**
5	99.93%	67.84%	81.04%	82.16%
6	99.91%	75.17%	85.34%	86.53%

Results NHSMM

Results HSMM without clustering, $M = 10$ and $k = 4039$

D	*Sensitivity*	*Specificity*	*Accuracy*	*G-mean*
3	75.21%	70.24%	72.29%	72.69%
4	75.12%	70.24%	72.25%	72.64%
5	75.25%	70.25%	72.30%	72.71%
6	**75.31%**	**70.23%**	**72.31%**	**72.72%**

Future Work

- To use current Deep Learning techniques such as **Long-short-term memory (LSTM)**, which do not explicitly model the time, but present a good performance in the classification of event log sequences.

Conclusions

- **Fail detection** in WfM/BPM systems can be carried out using HMM/HSMM/NHSMM models.
- The **performance** obtained by the HSMM model is superior to the one shown by the HMM and NHSMM.
- The best performance obtained in this work was an accuracy of **86.7%**.
- Experiments showed that setting ***M=10*** and ***D=4*** is a good choice.
- This is the first step to implement a system that allows predicting the behavior of the process in real time within a Big Data context.

References

- G. A. Fink, *Markov Models for Pattern Recognition*, vol. 1. London: Springer London, second edi ed., 2014.
- S.-Z. Yu, *Hidden Semi-Markov Models Theory, Algorithms and Applications.* Elsevier, oct 2016.
- W. M. P. Van der Aalst, “Business Process Management Demystified: A Tutorial on Models, Systems and Standards for Workflow Management,” *Lecture Notes in Computer Science: Lectures on Concurrency and Petri Nets: Advances in Petri Nets*, vol. 3098, pp. 1–65, 2004.

Questions

Thank you!