Incremental and adaptive learning for online monitoring of embedded software

Monica Loredana Angheloiu
Supervisors:
Marie-Odile Cordier
Laurence Rozé

Outline

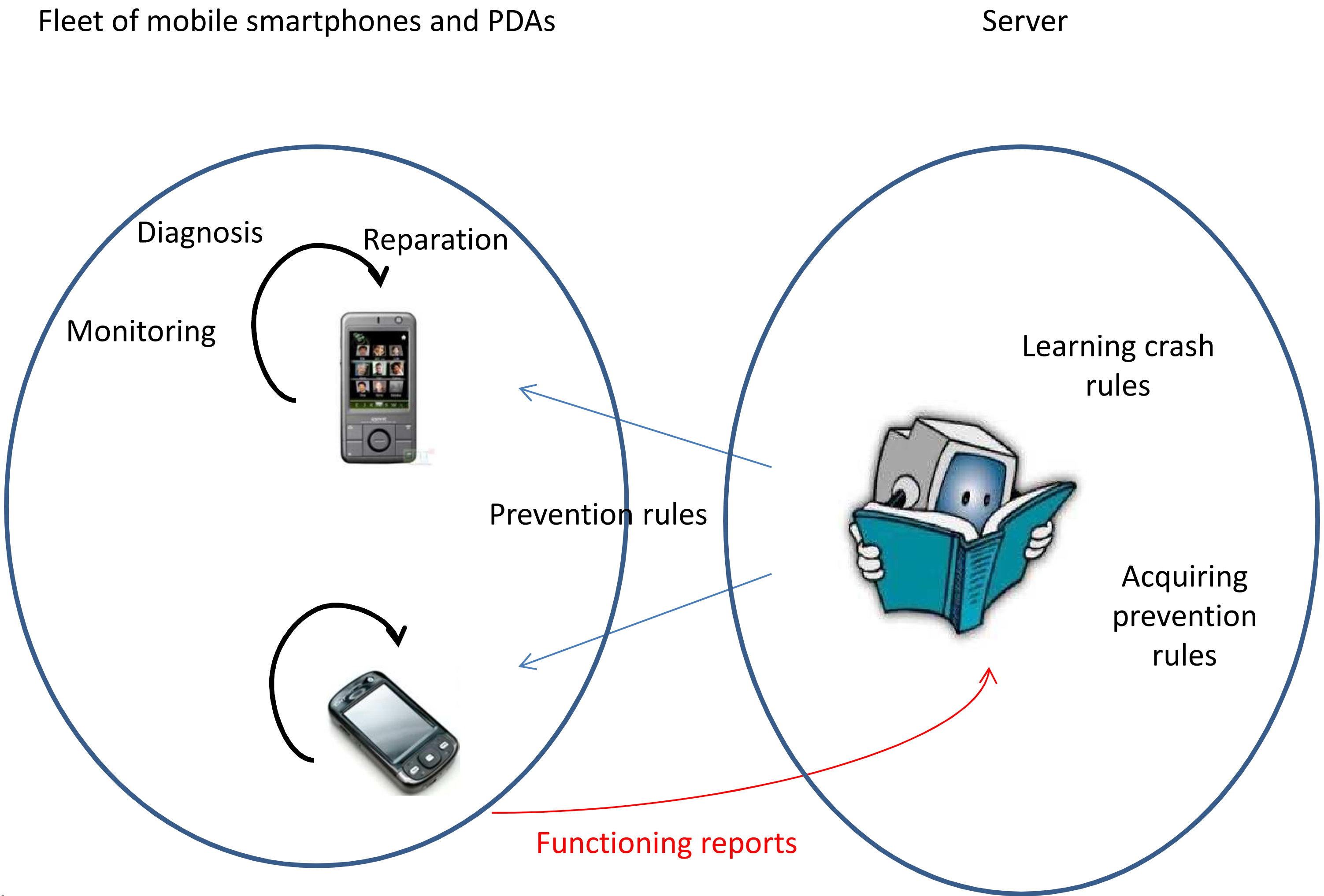
- Introduction
- Context of the internship
- Previous work
- Proposed approach
- Empirical Results
- Conclusion

Introduction

- Data is being collected continuously
- Useful information is "hidden"
- Human analysts can no longer infer knowledge

❖ The solution: machine learning

Context of the internship - Manage YourSelf



Problem statement

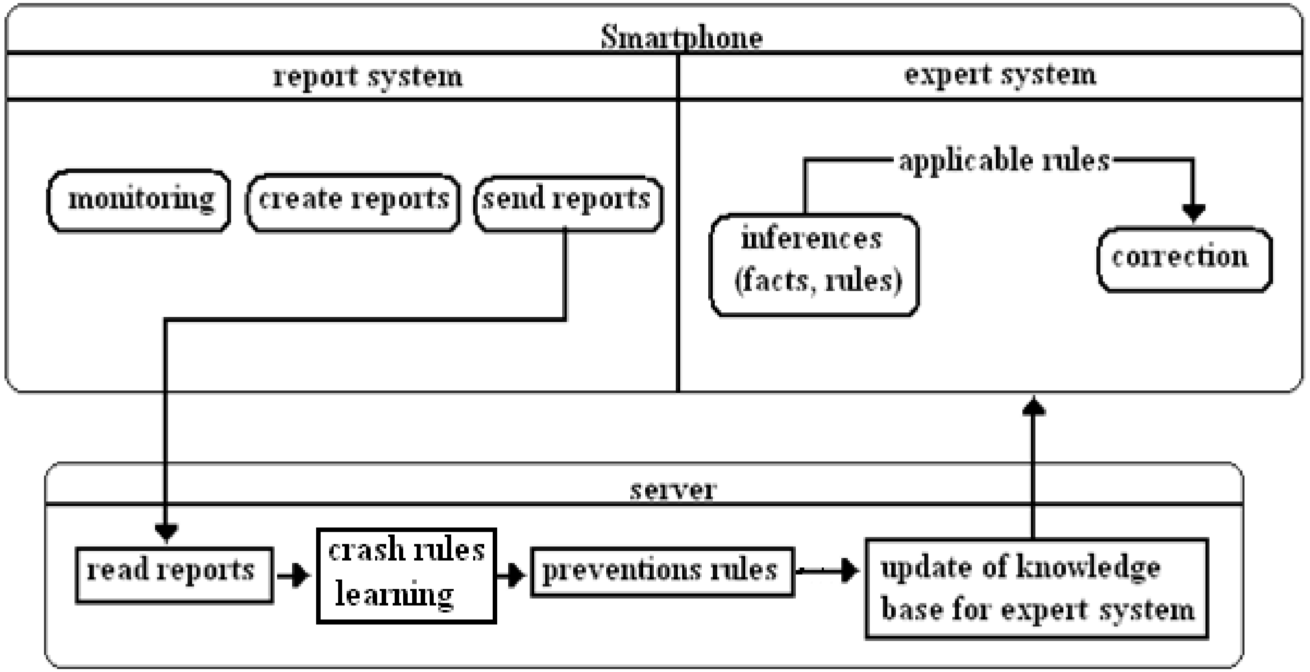
- Input data
  - Reports are generated by smartphones (or PDAs)


```xml
<rapport>
        <Date value="1302174135" />
        <Application name="Appli_Birds" value="running" />
        <OS value="Android 2.2" />
        <Battery value="1" />
        <Crash />
</rapport>
```


    - each time a problem appears
    - at regular time stamps in case of nominal behavior
  - Reports are sent in batch at a regular time stamps

- Objectives
  - Improve learning on server module using incremental learning

Manage YourSelf-general structure



Previous approach for server module:

- Decision trees are used for batch-learning
- All examples are stored

- batch learning vs. incremental learning
  at each learning step
  - batch-learning systems examine all examples one time
  - incremental systems examine the new training examples arrived

Challenges

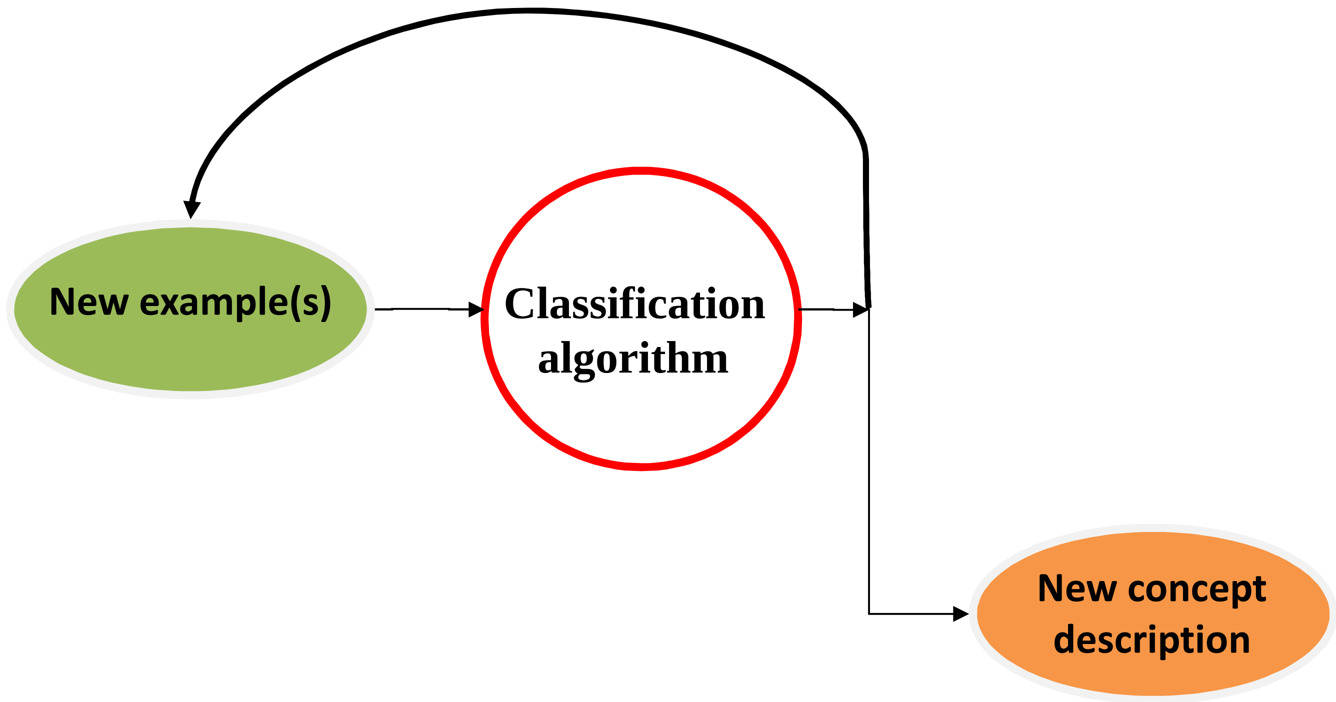
stabilize the processing time

reduce the storage requirements

detect concept drifts

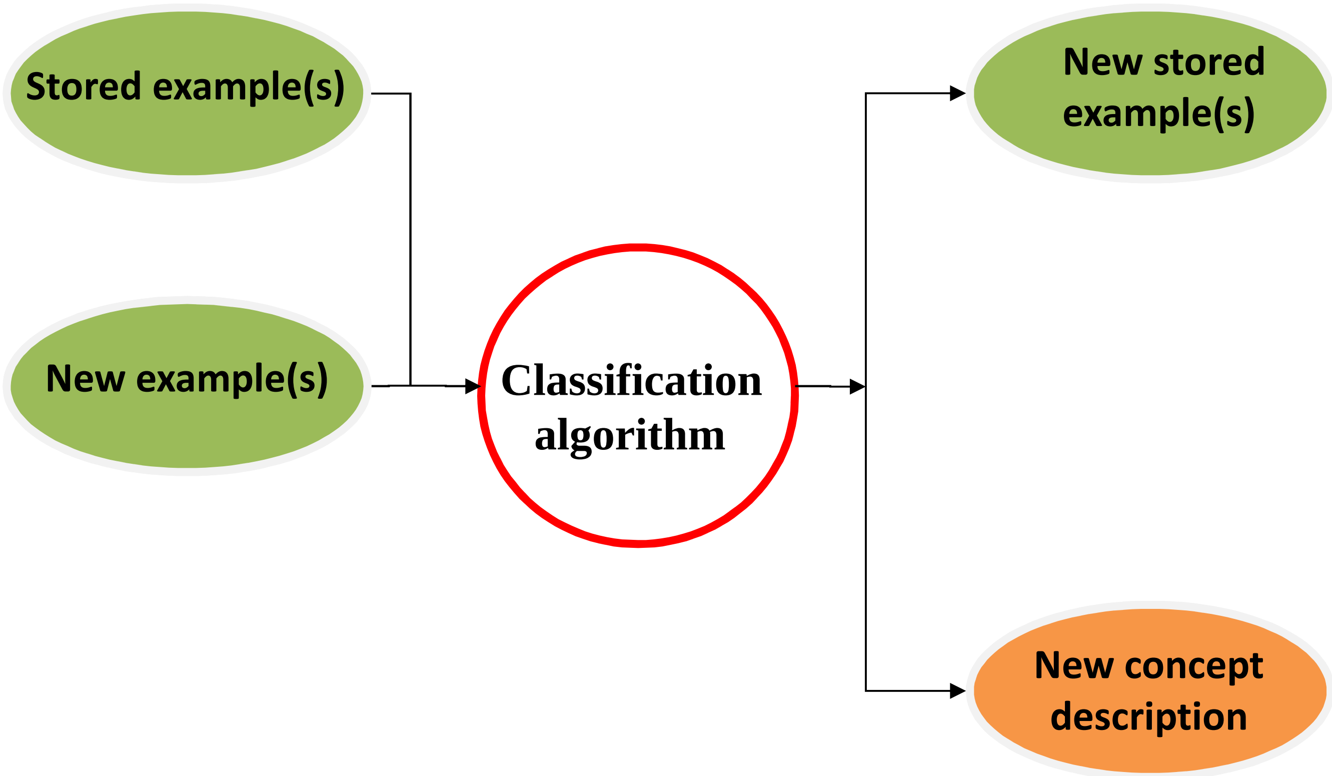
Few definitions

- Incremental learning



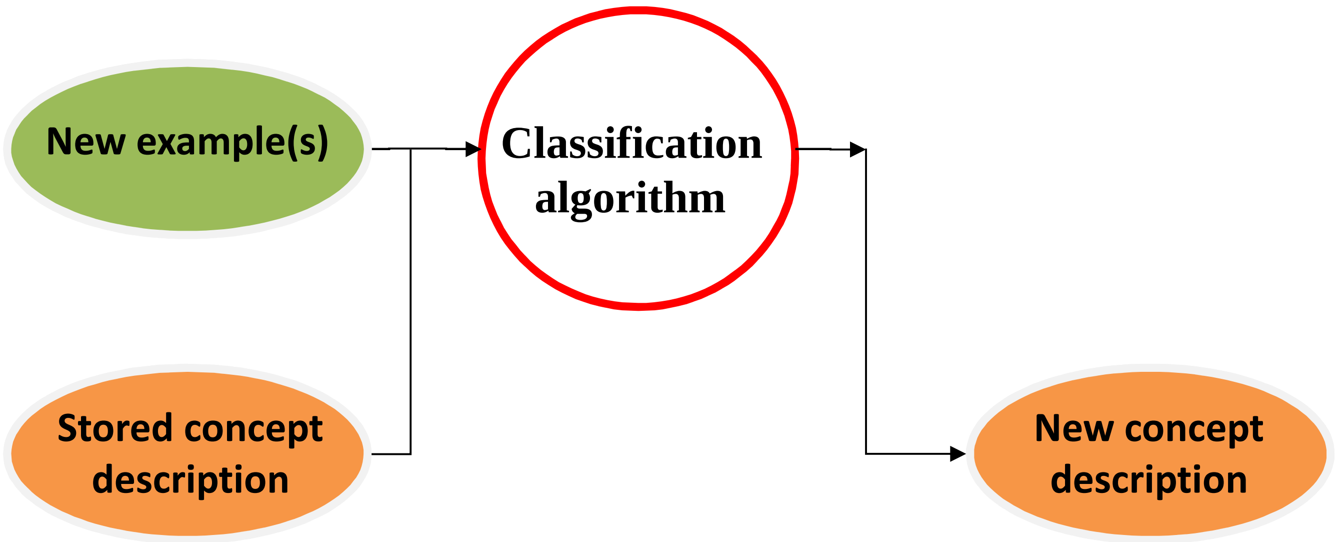
Few definitions

- Incremental learning
- Instance memory



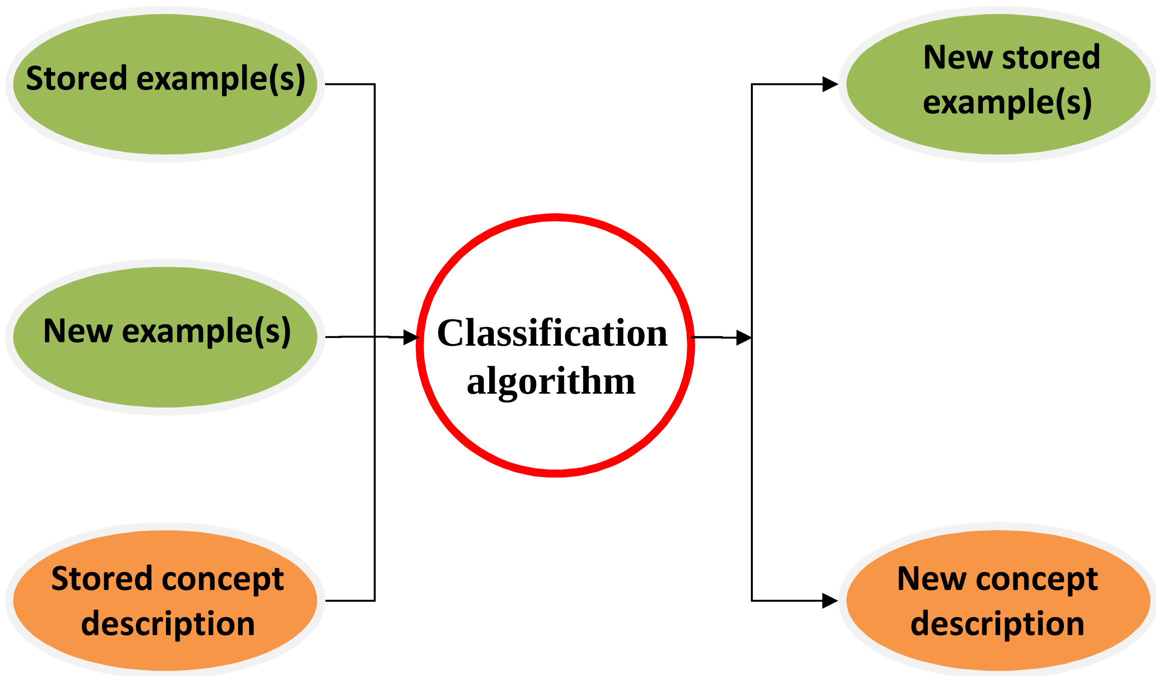
Few definitions

- Incremental learning
- Instance memory
- Concept memory



Few definitions

- Incremental learning
- Instance memory
- Concept memory



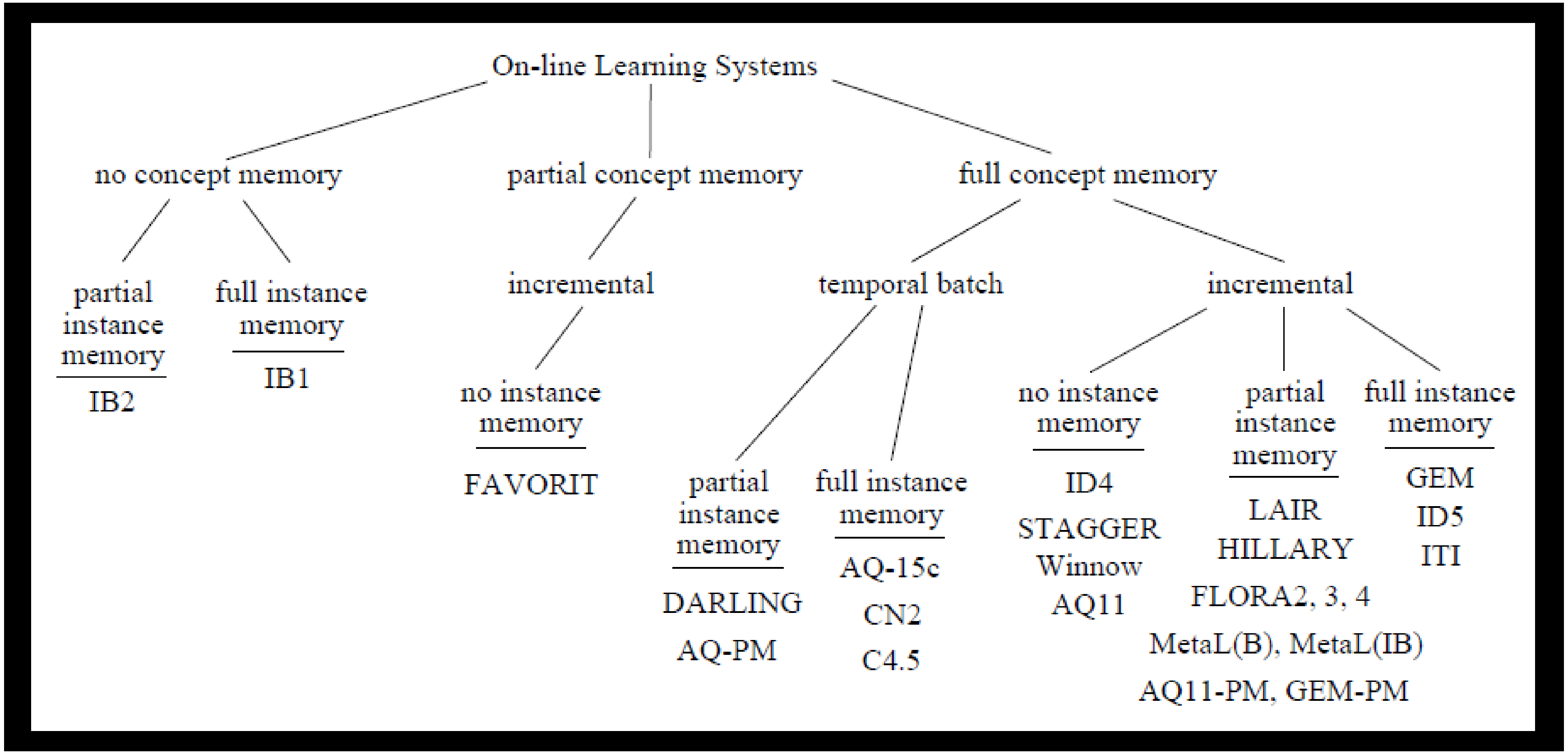
Few definitions

- Incremental learning
- Instance memory
- Concept memory
- Online learning
  - Incremental learning
  - Real time processing
  - Incoming order
  - Drift detection

Few definitions

- Incremental learning
- Concept memory
- Instance memory
- Online learning
- Concept drift
  - Hidden context changes

Representative approaches

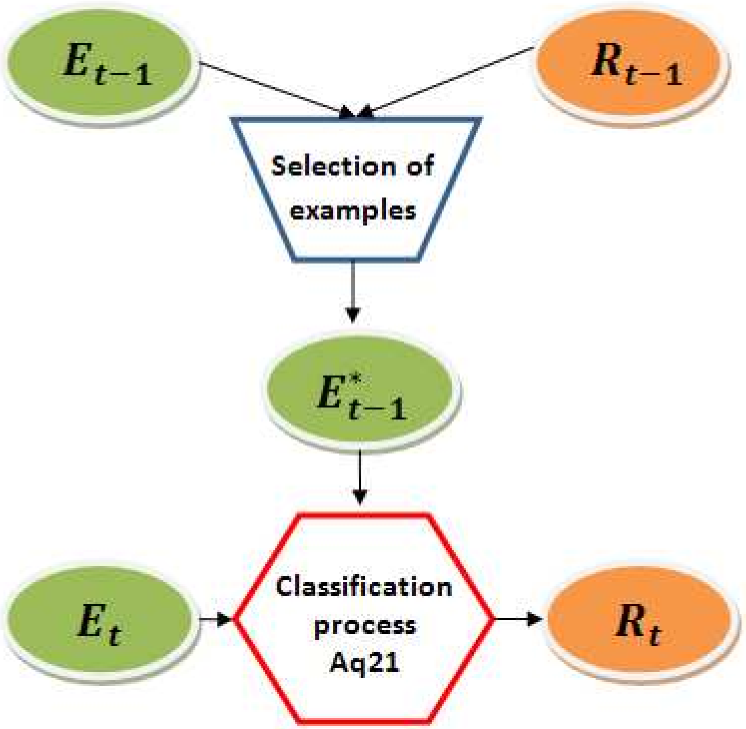


According to Maloof et all,2004[1]

A comparison between the representative incremental methods with partial instance memory

	AQ11-PM	FLORA	IB	FACIL	DARLING
Algorithm	Generalize training examples maximally	Generalize training examples when needed	Derived from nearest neighbor and do not generalize training examples	Derived from AQ11-PM	Make classification tree
Examples	Store only positive extreme examples	Store examples over a window of time	Store specific examples	Store both positive and negative examples, not necessarily extreme	Store specific neighboring examples
Interesting features		Keep old stable concepts	Use similarity function	Use the growth of a rule	Use a specific weight forgetting mechanism
Disadvantages	May cause overtraining	May delete available concept description	Computationally expensive	Has user defined parameters hard to tune	May not delete outdated concepts

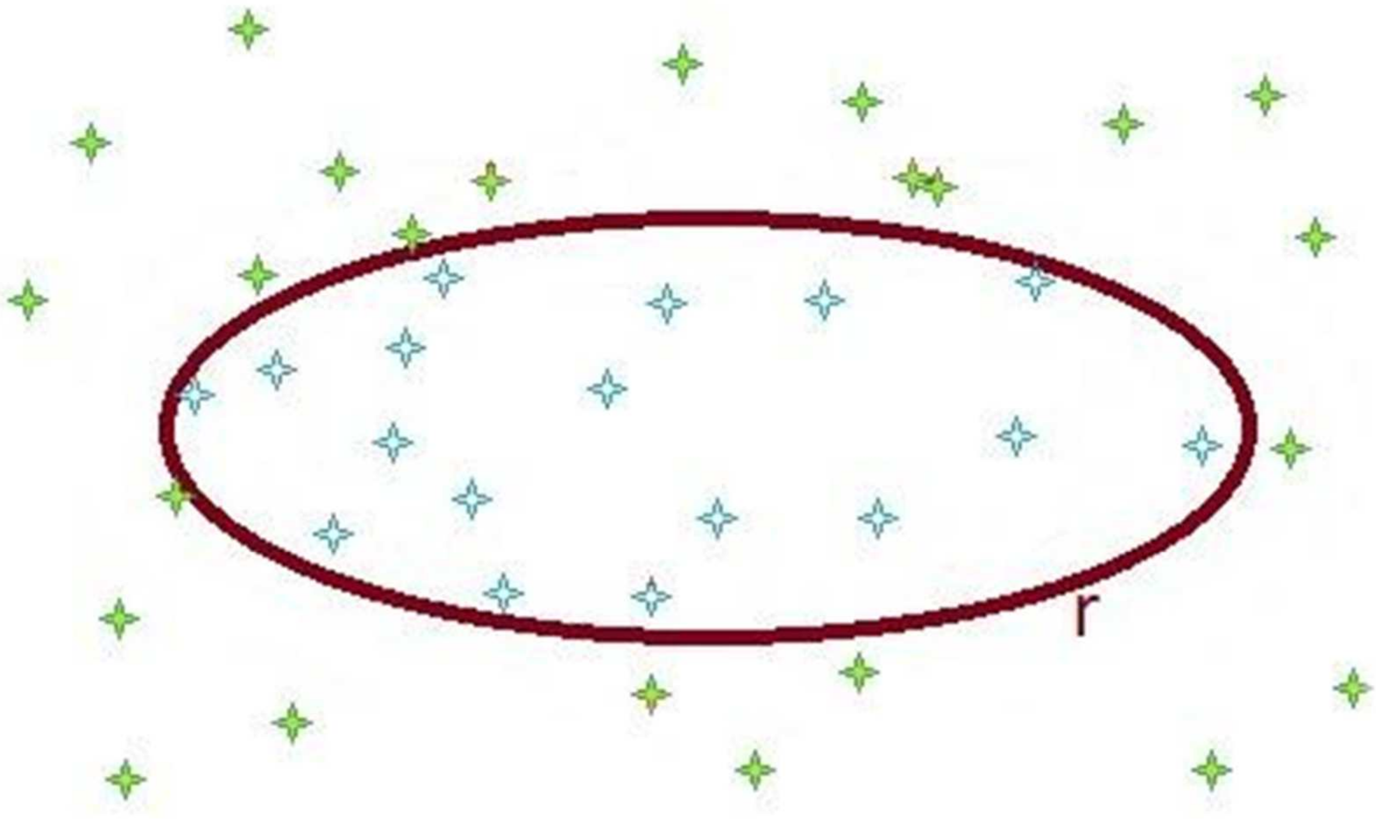
Proposed approach of incremental learning with partial instance memory and no concept memory



- **Classification basis non-incremental algorithm: AQ21**
- **Selection and storage of band border examples: Similar to AQ11-PM**

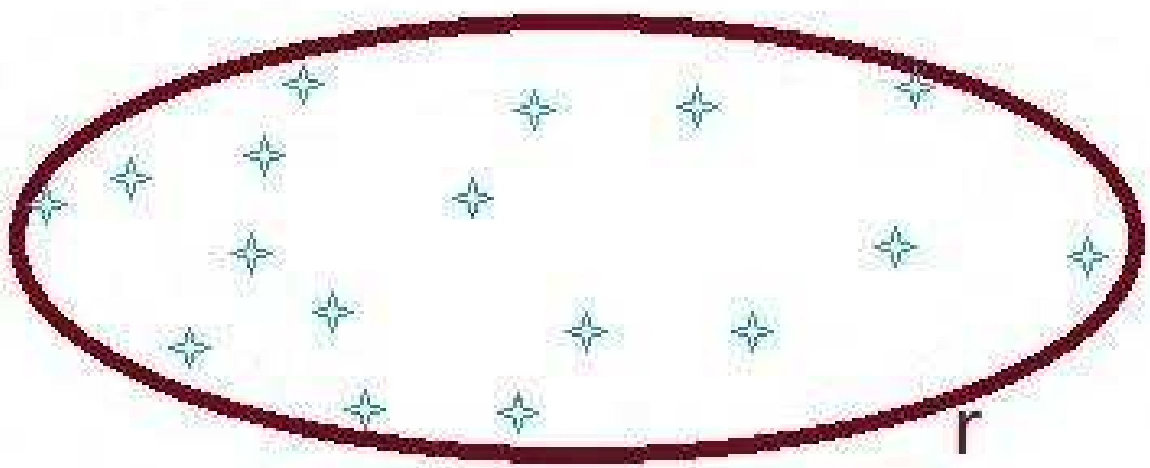
- Partial instance memory
  - the memory requirement is decreased and limited
  - the learning time is diminished
- Rule induction (AQ family)
  - the whole search space is not analyzed
  - the rules are quickly created and deleted
- Forgetting mechanism
  - the concept drifts are detected

Selection of examples



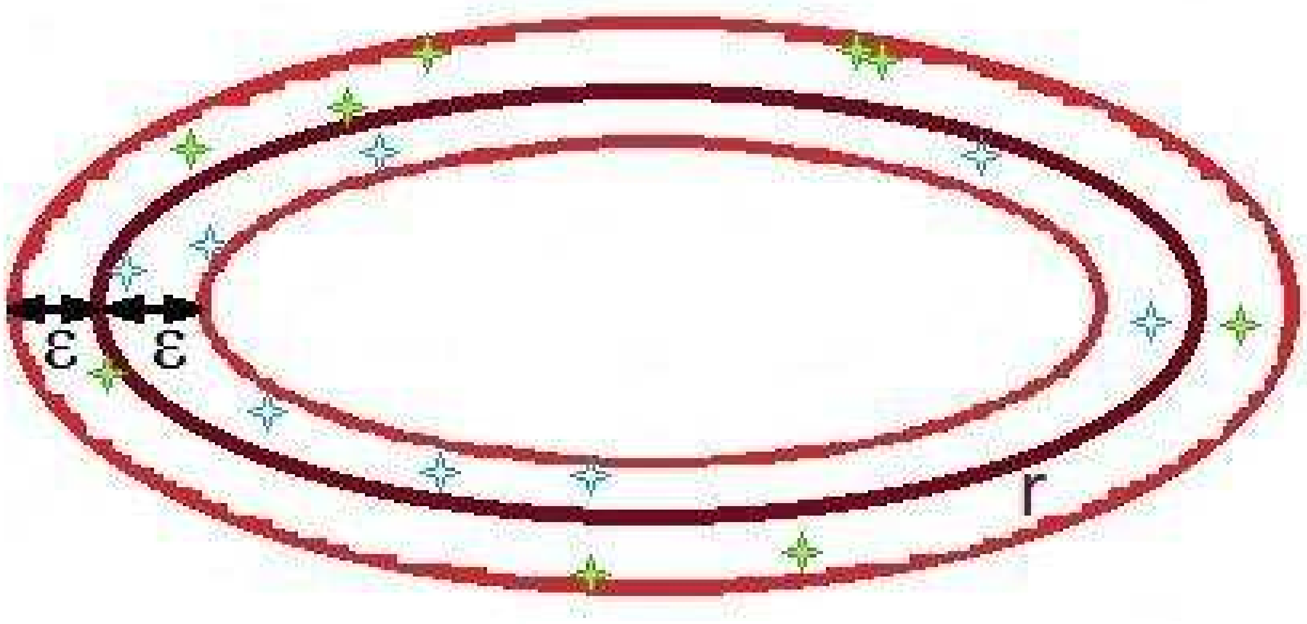
Selection of examples

- keep all examples of rules covering less than θ examples



Selection of examples

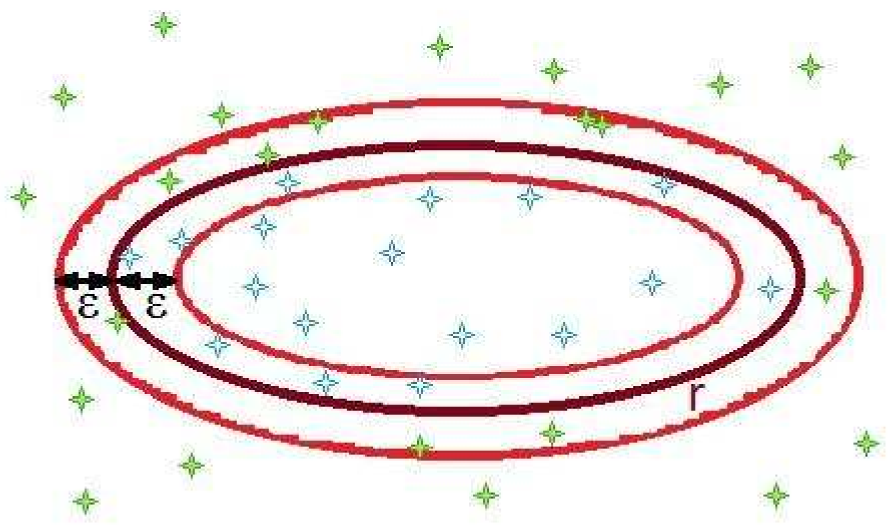
- keep band border examples, with the distance < ε, for rules covering over θ examples (similar to AQ11-PM)
- fix a maximum of stored examples (Ki for positive and Ke for negative)



- Distance function: $dist(r,e) = \sum_{i=1}^{m} d(c_i, x_i)$ $\quad d(c_i, x_i) = \begin{cases} \min\{(c_{iu} - x_i), (c_{il} - x_i)\}\ normalized, if\ c_i \in N \\ \neg(x_i \in c_i), if\ c_i \in S \end{cases}$

Selection of examples

- ❖ keep all examples of rules covering less than θ examples
- ❖ keep band border examples, with the distance < ε, for rules covering over θ examples (similar to AQ11-PM)
- ❖ fix a maximum of stored examples (Ki for positive and Ke for negative)
- ❖ age forgetting mechanism (when needed)



o Distance function: $dist(r,e) = \sum_{i=1}^{m} d(c_i, x_i)$ $d(c_i, x_i) = \begin{cases} \min\{(c_{iu} - x_i), (c_{il} - x_i)\}\ normalized, if\ c_i \in N \\ \neg(x_i \in c_i), if\ c_i \in S \end{cases}$

What we want to assess by experimentations

- Non-incremental
    - Get results similar to behavior rules used for simulation
- Incremental in stationary environment
    - Keep the memory limited
    - Keep learning time almost constant
    - Get results similar to the non-incremental approach
- Incremental in a drifting environment
    - Achieve rules similar to behavior rules of current iteration
    - Detect and track drifts using a forgetting mechanism

Behavior rules for simulating input data

A total of 11 rules:

- General
  - If battery < 3% then crash low battery
  - If memory RAM > 95% then crash memory full
  - If memory ROM > ROM size – 1000 MB then crash memory full
- Operation system
  - If Appli_Incompatible_Android open and OS =Android then crash applicrash
  - If Appli_Incompatible_IOS open and OS =IOS then crash application
  - If Appli_Incompatible_MWP open and OS = MWP then crash application
- Brand
  - If brand=Sony and battery < 8% then crash low battery
  - If brand =Apple and Appli _GPS open and Appli_Incompatible_GPS open then crash application
- Model
  - If model = Omnia7 and Appli_Incompatible_Omnia open then crash application
  - If model =GalaxyMini and Appli _GSM open and Appli _WIFI open and Appli _GPS open and battery< 10% then crash low battery
- Specific
  - If Appli_Incompatible_Telephone open then crash application

Experimentations

❖ Non-incremental

Learning time	Total positive	Total negative	Total stored examples	Number of important rules	Total number of rules	Precision	Recall
60 m	11,211	145,757	156,968	12	24	100%	99.67%

❖ Incremental in a a stationary environment

- 6 steps of incremental learning
- input of one step include:
  - approximately 30.800 new incoming reports
    - » 88 different smartphones
  - approximately a 5 days simulation / smartphone
    - » 350 reports / smartphone

Empirical results

No	θ	ε	Ki	Ke	Time	Time last incremental step	Mean positive	Mean negative	Mean of stored examples	Number of important rules	Total number of rules	Precision	Recall
-	-	-	-	-	-	60 m	11,211	145,757	156,968	12	24	100%	99.67%
1	25	0	250	250	111 m 20 s	19 m 38 s	1,229.6	904.5	32,309.4	11	41	56.47%	93.22%
2	25	0.5	250	250	143 m 29 s	31 m 35 s	2,359.4	3,859.5	35,315.4	24	62	99.67%	95.07%
3	25	1	250	250	131 m 33 s	27 m 12 s	2341	5616	36,527.6	25	62	99.82%	96.75%
4	25	1.5	250	250	115 m 37 s	18 m 58 s	2,297.8	5,981.6	36,643.6	23	61	99.94%	96.47%
5	25	2	250	250	119 m 10 s	24 m 22 s	2,279.3	5,997.3	36,643.6	24	64	99.94%	96.47%
6	25	2.5	250	250	122 m 23 s	29 m 1 s	2,384.2	6,291.8	36,676.8	27	68	99.86%	96.56%
7	30	0	100	100	97 m 40 s	11 m 19 s	880.8	499.8	31,857.6	11	42	98.77%	93.66%
8	30	0.5	100	100	106 m 52 s	18 m 27 s	1,467.8	1,984.8	33,256.4	17	57	99.00%	96.29%
9	30	1	100	100	106 m 1 s	24 m 8 s	1,457.8	2,644.1	33,809.6	19	60	99.65%	95.29%
10	30	1.5	100	100	104 m 6 s	25 m 4 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
11	30	2	100	100	101 m 18 s	24 m 16 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
12	30	2.5	100	100	101 m 5 s	24 m 12 s	1,437.3	2,895.3	33,845.6	18	68	99.71%	93.28%
						23 m 11 s	1,855.3	3,753.3	34,782.2	20	57		

Empirical results

No	θ	ε	Ki	Ke	Time	Time last incremental step	Mean positive	Mean negative	Mean of stored examples	Number of important rules	Total number of rules	Precision	Recall
-	-	-	-	-	-	60 m	11,211	145,757	156,968	12	24	100%	99.67%
1	25	0	250	250	111 m 20 s	19 m 38 s	1,229.6	904.5	32,309.4	11	41	56.47%	93.22%
2	25	0.5	250	250	143 m 29 s	31 m 35 s	2,359.4	3,859.5	35,315.4	24	62	99.67%	95.07%
3	25	1	250	250	131 m 33 s	27 m 12 s	2341	5616	36,527.6	25	62	99.82%	96.75%
4	25	1.5	250	250	115 m 37 s	18 m 58 s	2,297.8	5,981.6	36,643.6	23	61	99.94%	96.47%
5	25	2	250	250	119 m 10 s	24 m 22 s	2,279.3	5,997.3	36,643.6	24	64	99.94%	96.47%
6	25	2.5	250	250	122 m 23 s	29 m 1 s	2,384.2	6,291.8	36,676.8	27	68	99.86%	96.56%
7	30	0	100	100	97 m 40 s	11 m 19 s	880.8	499.8	31,857.6	11	42	98.77%	93.66%
8	30	0.5	100	100	106 m 52 s	18 m 27 s	1,467.8	1,984.8	33,256.4	17	57	99.00%	96.29%
9	30	1	100	100	106 m 1 s	24 m 8 s	1,457.8	2,644.1	33,809.6	19	60	99.65%	95.29%
10	30	1.5	100	100	104 m 6 s	25 m 4 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
11	30	2	100	100	101 m 18 s	24 m 16 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
12	30	2.5	100	100	101 m 5 s	24 m 12 s	1,437.3	2,895.3	33,845.6	18	68	99.71%	93.28%
						23 m 11 s	1,855.3	3,753.3	34,782.2	20	57		

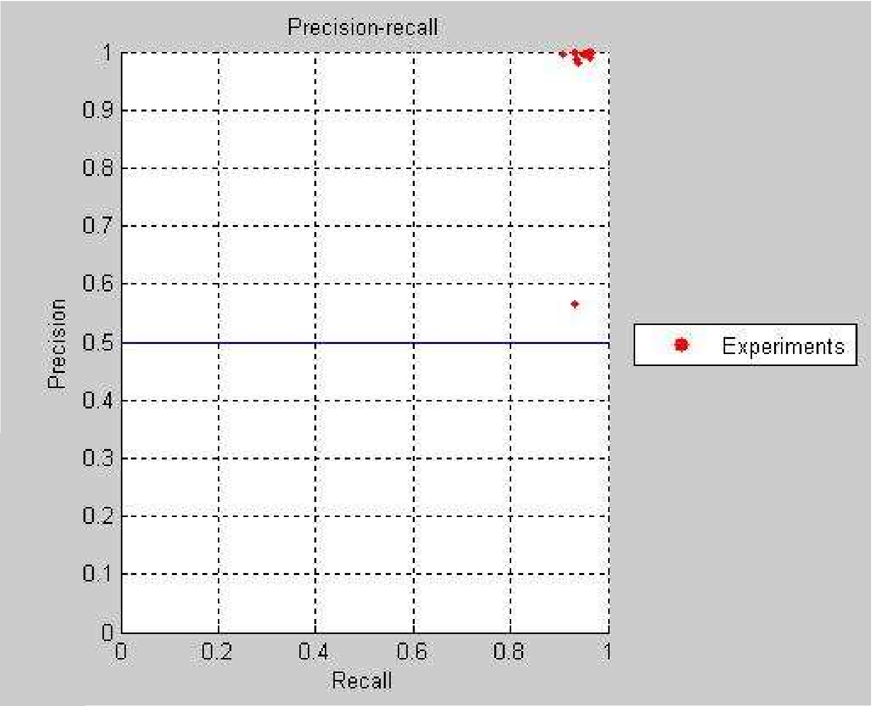
Empirical results

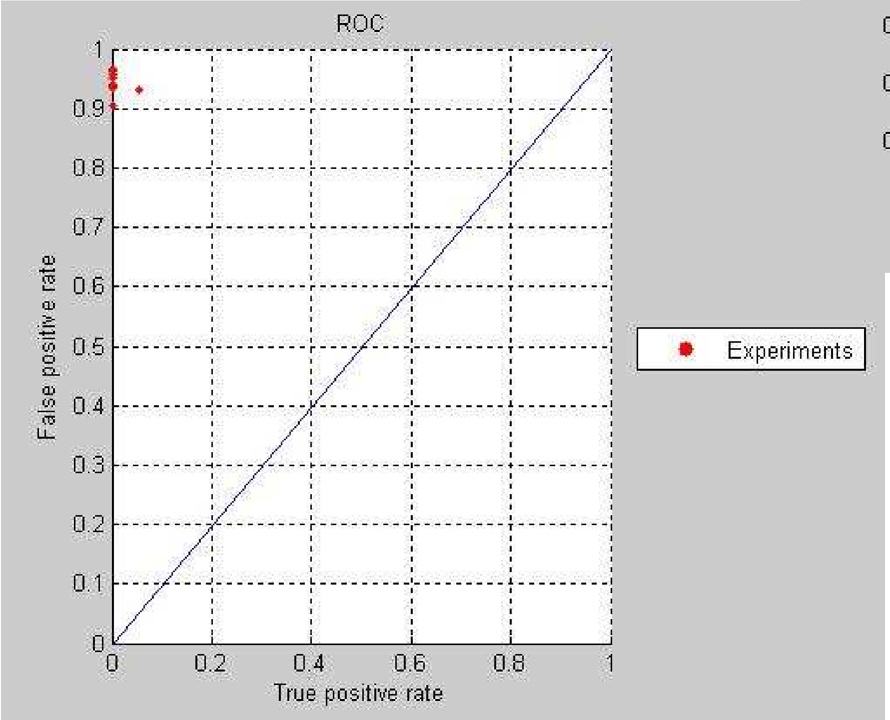
No	θ	ε	Ki	Ke	Time	Time last incremental step	Mean positive	Mean negative	Mean of stored examples	Number of important rules	Total number of rules	Precision	Recall
-	-	-	-	-	-	60 m	11,211	145,757	156,968	12	24	100%	99.67%
1	25	0	250	250	111 m 20 s	19 m 38 s	1,229.6	904.5	32,309.4	11	41	56.47%	93.22%
2	25	0.5	250	250	143 m 29 s	31 m 35 s	2,359.4	3,859.5	35,315.4	24	62	99.67%	95.07%
3	25	1	250	250	131 m 33 s	27 m 12 s	2341	5616	36,527.6	25	62	99.82%	96.75%
4	25	1.5	250	250	115 m 37 s	18 m 58 s	2,297.8	5,981.6	36,643.6	23	61	99.94%	96.47%
5	25	2	250	250	119 m 10 s	24 m 22 s	2,279.3	5,997.3	36,643.6	24	64	99.94%	96.47%
6	25	2.5	250	250	122 m 23 s	29 m 1 s	2,384.2	6,291.8	36,676.8	27	68	99.86%	96.56%
7	30	0	100	100	97 m 40 s	11 m 19 s	880.8	499.8	31,857.6	11	42	98.77%	93.66%
8	30	0.5	100	100	106 m 52 s	18 m 27 s	1,467.8	1,984.8	33,256.4	17	57	99.00%	96.29%
9	30	1	100	100	106 m 1 s	24 m 8 s	1,457.8	2,644.1	33,809.6	19	60	99.65%	95.29%
10	30	1.5	100	100	104 m 6 s	25 m 4 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
11	30	2	100	100	101 m 18 s	24 m 16 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
12	30	2.5	100	100	101 m 5 s	24 m 12 s	1,437.3	2,895.3	33,845.6	18	68	99.71%	93.28%
						23 m 11 s	1,855.3	3,753.3	34,782.2	20	57		

Empirical results

No	θ	ε	Ki	Ke	Time	Time last incremental step	Mean positive	Mean negative	Mean of stored examples	Number of important rules	Total number of rules	Precision	Recall
-	-	-	-	-	-	60 m	11,211	145,757	156,968	12	24	100%	99.67%
1	25	0	250	250	111 m 20 s	19 m 38 s	1,229.6	904.5	32,309.4	11	41	56.47%	93.22%
2	25	0.5	250	250	143 m 29 s	31 m 35 s	2,359.4	3,859.5	35,315.4	24	62	99.67%	95.07%
3	25	1	250	250	131 m 33 s	27 m 12 s	2341	5616	36,527.6	25	62	99.82%	96.75%
4	25	1.5	250	250	115 m 37 s	18 m 58 s	2,297.8	5,981.6	36,643.6	23	61	99.94%	96.47%
5	25	2	250	250	119 m 10 s	24 m 22 s	2,279.3	5,997.3	36,643.6	24	64	99.94%	96.47%
6	25	2.5	250	250	122 m 23 s	29 m 1 s	2,384.2	6,291.8	36,676.8	27	68	99.86%	96.56%
7	30	0	100	100	97 m 40 s	11 m 19 s	880.8	499.8	31,857.6	11	42	98.77%	93.66%
8	30	0.5	100	100	106 m 52 s	18 m 27 s	1,467.8	1,984.8	33,256.4	17	57	99.00%	96.29%
9	30	1	100	100	106 m 1 s	24 m 8 s	1,457.8	2,644.1	33,809.6	19	60	99.65%	95.29%
10	30	1.5	100	100	104 m 6 s	25 m 4 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
11	30	2	100	100	101 m 18 s	24 m 16 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
12	30	2.5	100	100	101 m 5 s	24 m 12 s	1,437.3	2,895.3	33,845.6	18	68	99.71%	93.28%
						23 m 11 s	1,855.3	3,753.3	34,782.2	20	57		

Empirical results





The overtraining impact

- The rules achieved incrementally have a good precision-recall, but are not as general as those retrieved in non-incremental, because of border examples stored
  - The non-incremental rule:
    - (batterie<= 7) and (brand = 'Sony' or brand = 'Nokia') and (numero<= 50000000)
  - The incremental set of 12 rules:
    - (batterie between 4 and 7) and (brand = 'Sony') and ("app_Appli_GPS" = 'running')
    - (batterie <= 7) and (brand = 'Sony') and (memoirephysique >= 677) and ("app_Appli_WIFI" = 'running')
    - (batterie <= 7) and (brand = 'Sony' or brand = 'Apple') and (memoirephysique <= 1400) and ("app_Appli_Call" = 'running')
    - (batterie between 4 and 7) and (brand = 'Sony' or brand = 'Apple') and (memoirephysique >= 1402) and ("app_Appli_Call" = 'running')
    - (batterie <= 7) and (brand = 'Sony' or brand = 'Apple') and (memoirephysique between 1402 and 1872) and ("app_Appli_Call" = 'running')
    - …

Experimentations in a drifting environment

- The fleet of smarphones are replaced each time we recomputed concept description and we pass from a smartphone model to another at each incremental step
- The age forgetting mechanism is used
  - 8 steps of incremental learning
  - input of one step include:
    - approximately 20.000 reports
    - all reports generated for one smartphone model
      - » **Models:** GalaxyMini, Galaxy S2, IPhone 4S, Omnia 7,
        Lumia 900, Lumia 800, Xperia Pro, Xperia Mini
    - each model includes 11 different smartphones simulations, during a month

o In experiments the age parameter of the forgetting mechanism is set to 3

Empirical results

Experiment 1: Lumia 800

Models order: GalaxyMini, Galaxy S2, Xperia Pro, Xperia Mini,
IPhone 4S, Omnia 7, Lumia 900, Lumia 800

- Behavior rules for the last iteration
    - (batterie < 3)
    - (memory RAM > 950)
    - (memory ROM > 15000)
    - ("app_Appli_Incompatible_MicrosoftWindowsPhone" = 'running')
    - ("app_Appli_Incompatible_Telephone" = 'running')

- Rules achieved
    - (batterie <= 2)
    - (memoirevive >= 922)
    - ("app_Appli_Incompatible_MicrosoftWindowsPhone" = 'running')
    - ("app_Appli_Incompatible_Telephone" = 'running')

Empirical results

Experiment 2: Xperia Mini

Models order: GalaxyMini, Galaxy S2, IPhone 4S, Omnia 7,
Lumia 900, Lumia 800, Xperia Pro, Xperia Mini

- Behavior rules for the last iteration
    - (batterie < 3)
    - (memory RAM > 950)
    - (memory ROM > 3000)
    - (" app_Appli_Incompatible_Android " = 'running')
    - (batterie < 8) and (modelu = XperiaMini')
    - ("app_Appli_Incompatible_Telephone" = 'running')

- Rules achieved
    - (batterie <= 7) and (modelu <> 'Lumia800')
    - (memoirevive >= 933)
    - (batterie <= 2)
    - (memoirevive <= 543) and ("app_Appli_Incompatible_Android" = 'running')
    - (version <> 'Android23') and ("app_Appli_Incompatible_MicrosoftWindowsPhone" = 'running')
    - ("app_Appli_Incompatible_Telephone" = 'running')

Conclusions

❖ The proposed approach

- use simulated data
- limit the memory requirements
- keep learning time almost constant
- store band border examples (similar to AQ11-PM)
- incorporate a forgetting mechanism
- detect and adapt to concept drifts

Future work

- Improve feature selection
  - » drop irrelevant dimensions
- Develop a concept memory approach
  - » filter rules according to some specific parameters
  - » limit the number of saved rules for each iteration
  - » merge current rules with previous ones
- Deal with examples which are masked by prevention rules

Thank you !

Incremental and adaptive learning for online monitoring of embedded software

Monica Loredana Angheloiu
Supervisors:
Marie-Odile Cordier
Laurence Rozé

Outline

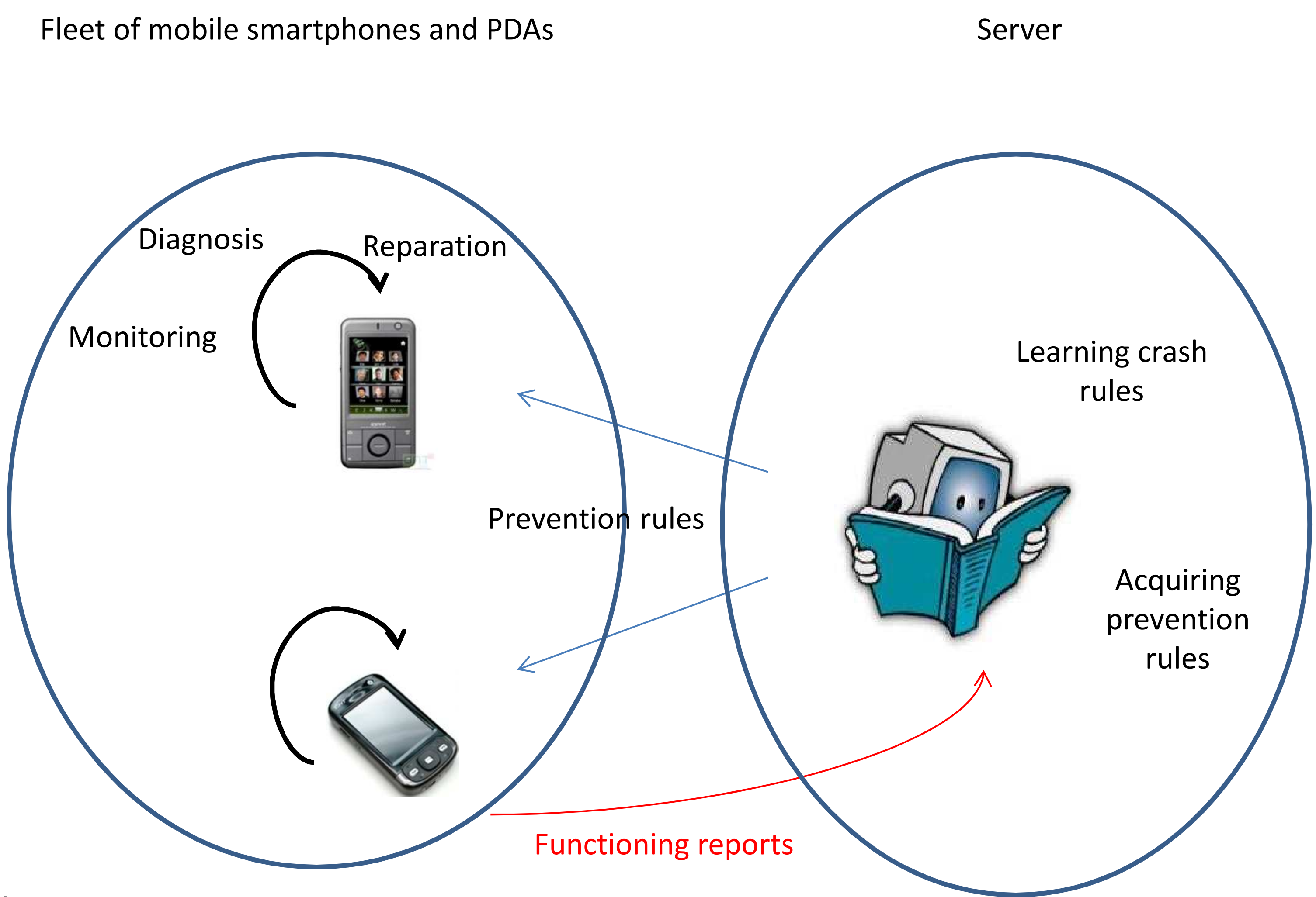
- Introduction
- Context of the internship
- Previous work
- Proposed approach
- Empirical Results
- Conclusion

Introduction

- Data is being collected continuously
- Useful information is “hidden”
- Human analysts can no longer infer knowledge

❖ The solution: machine learning

Context of the internship - Manage YourSelf



Problem statement

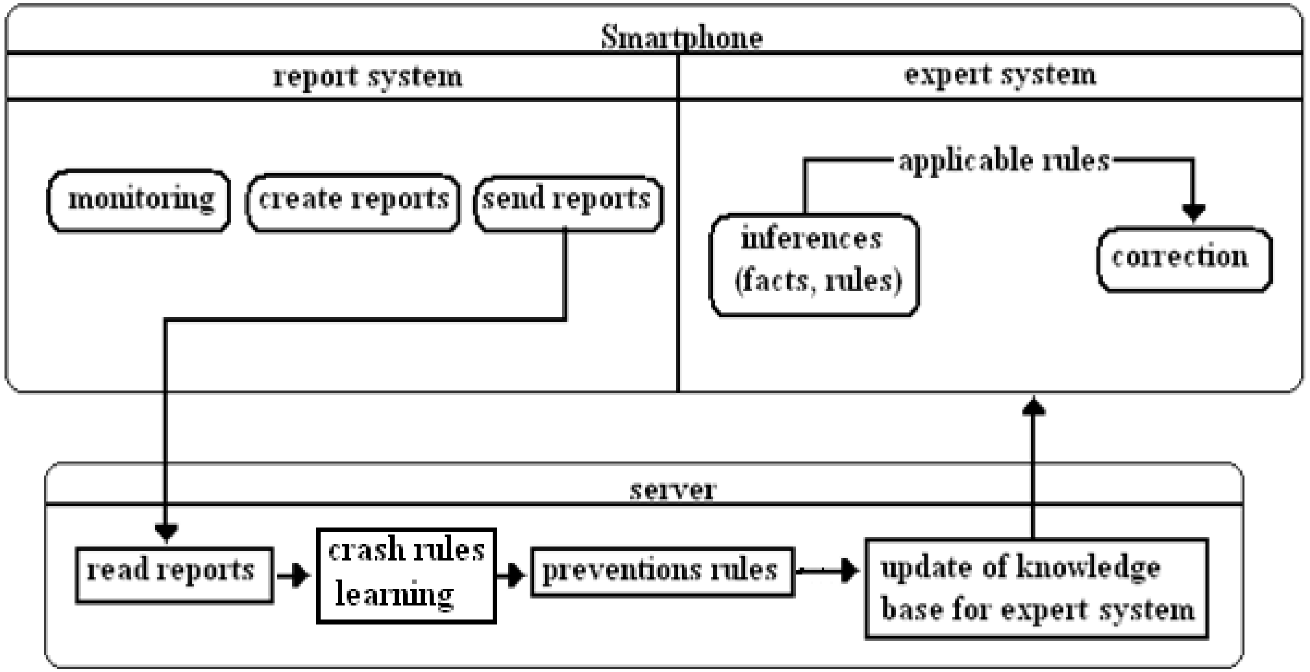
- Input data
  - Reports are generated by smartphones (or PDAs)


```xml
<rapport>
        <Date value="1302174135" />
        <Application name="Appli_Birds" value="running" />
        <OS value="Android 2.2" />
        <Battery value="1" />
        <Crash />
</rapport>
```


    - each time a problem appears
    - at regular time stamps in case of nominal behavior
  - Reports are sent in batch at a regular time stamps

- Objectives
  - Improve learning on server module using incremental learning

Manage YourSelf-general structure



Previous approach for server module:

- Decision trees are used for batch-learning
- All examples are stored

- batch learning vs. incremental learning
  at each learning step
  - batch-learning systems examine all examples one time
  - incremental systems examine the new training examples arrived

Challenges

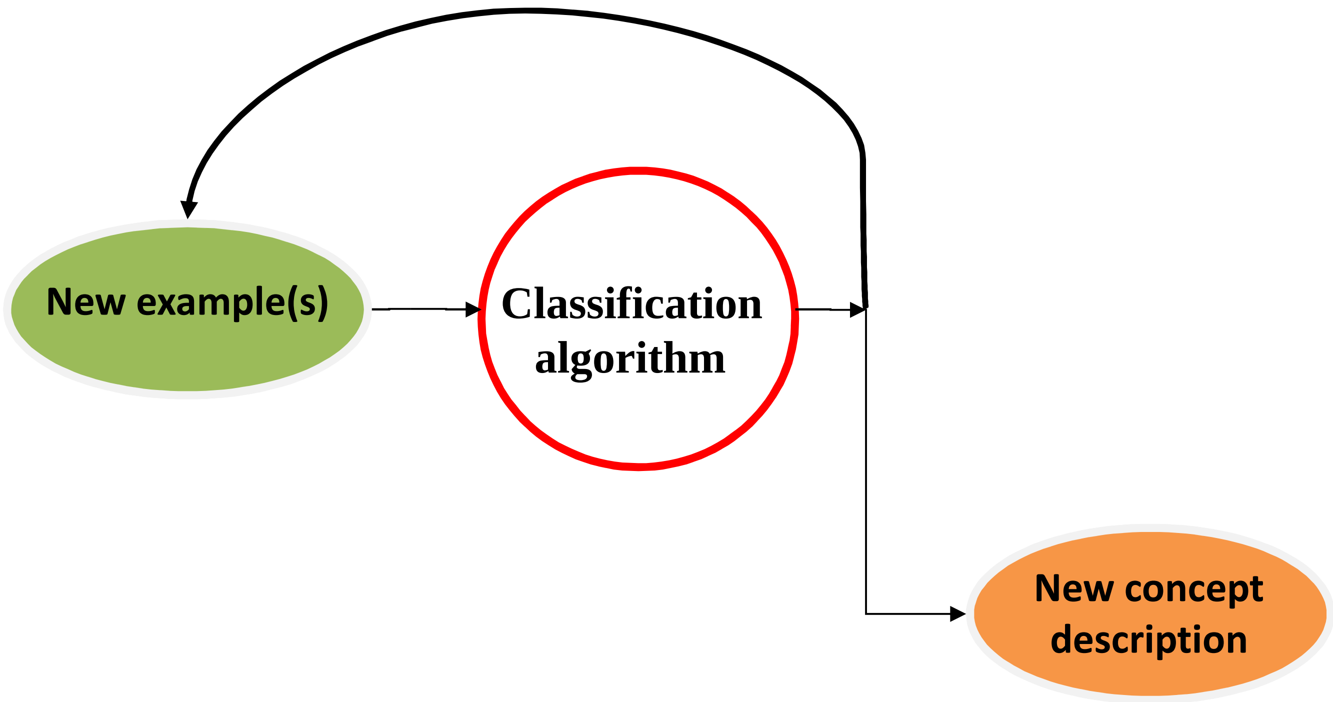
stabilize the processing time

reduce the storage requirements

detect concept drifts

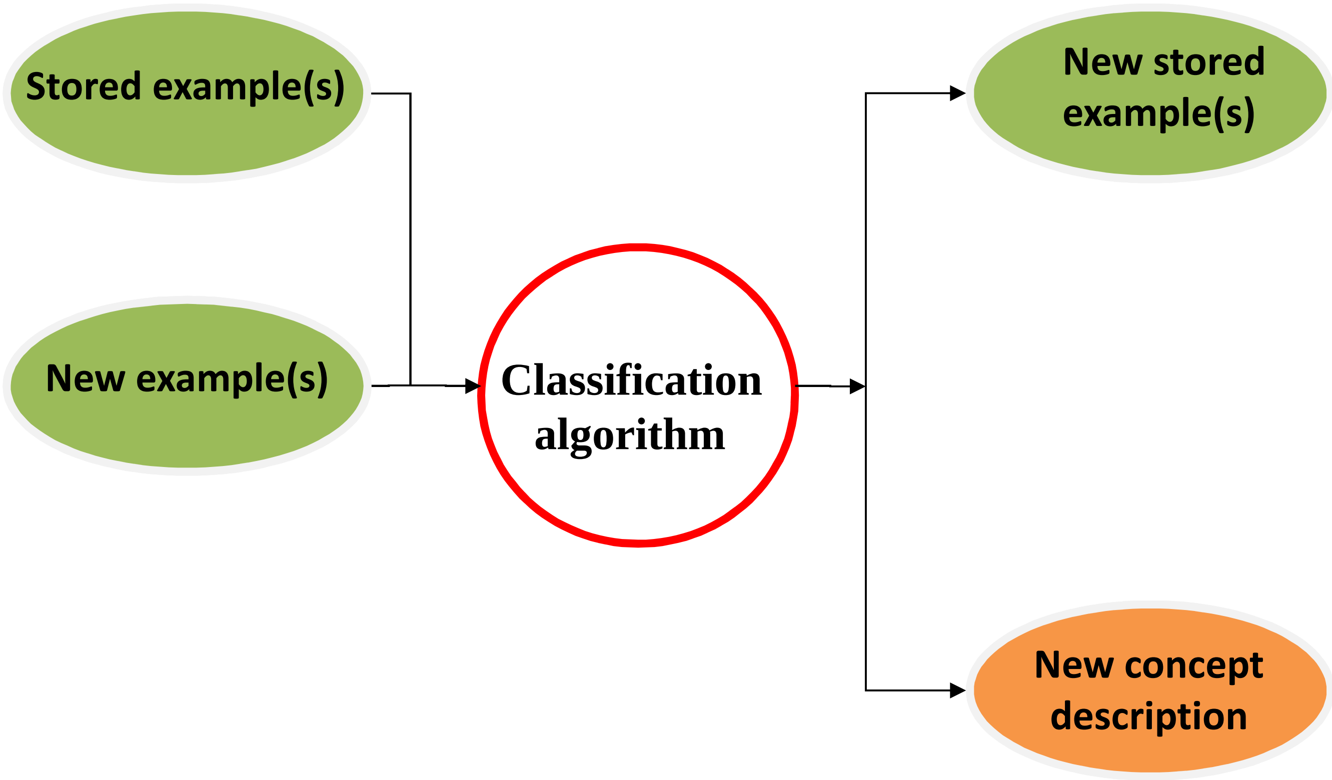
Few definitions

- Incremental learning



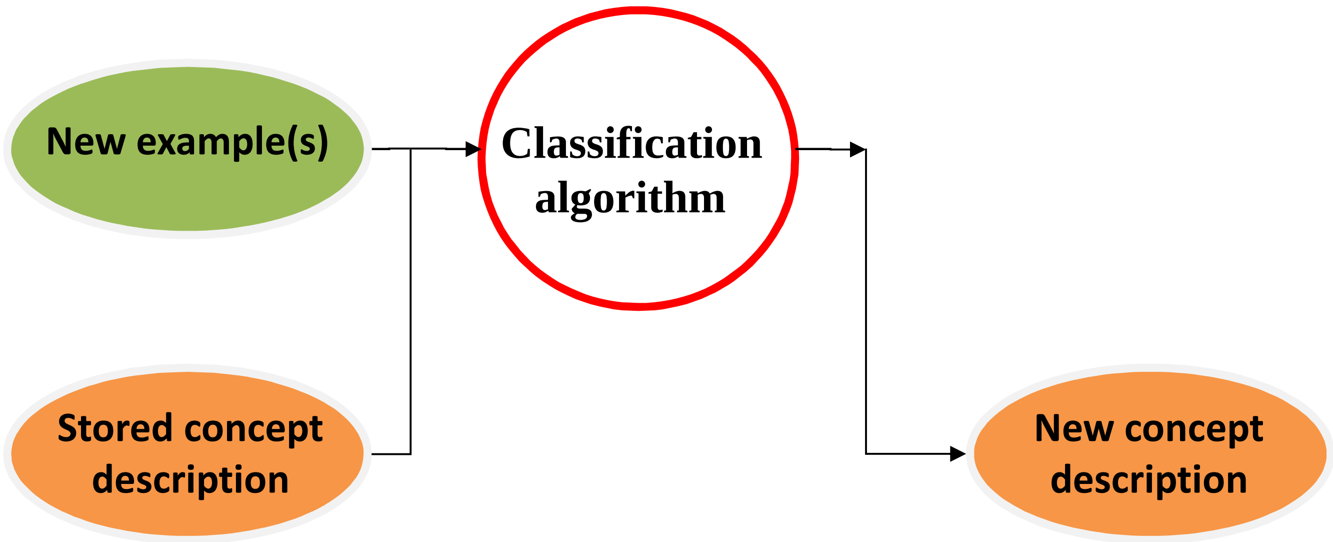
Few definitions

- Incremental learning
- Instance memory



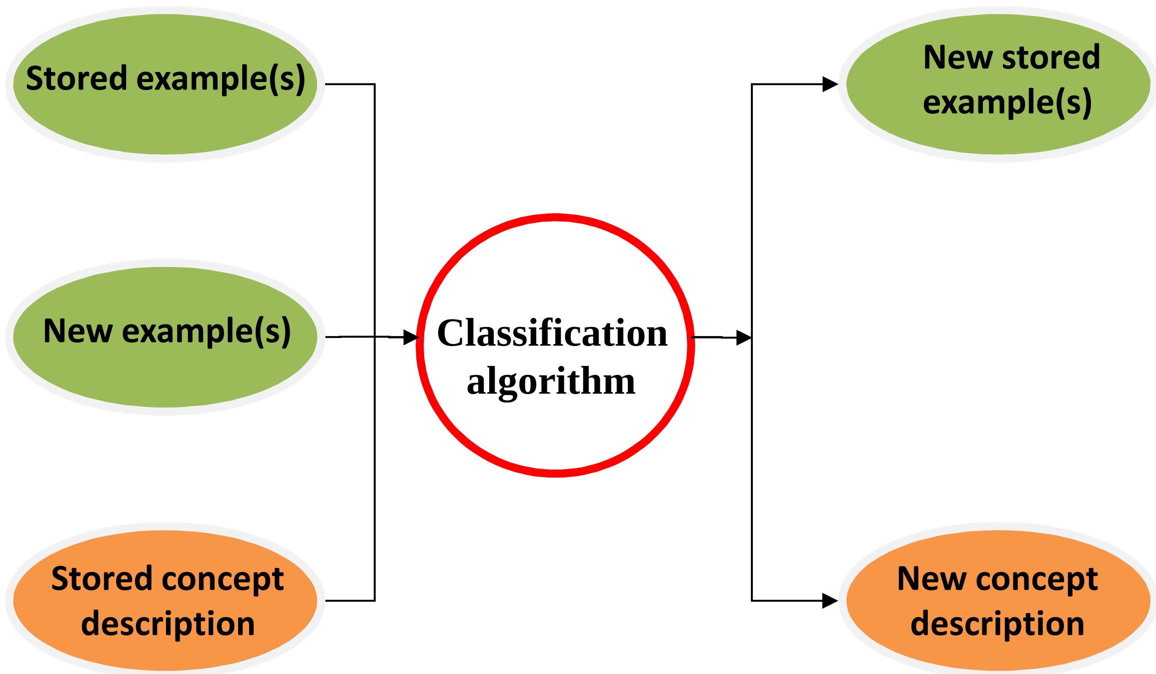
Few definitions

- Incremental learning
- Instance memory
- Concept memory



Few definitions

- Incremental learning
- Instance memory
- Concept memory



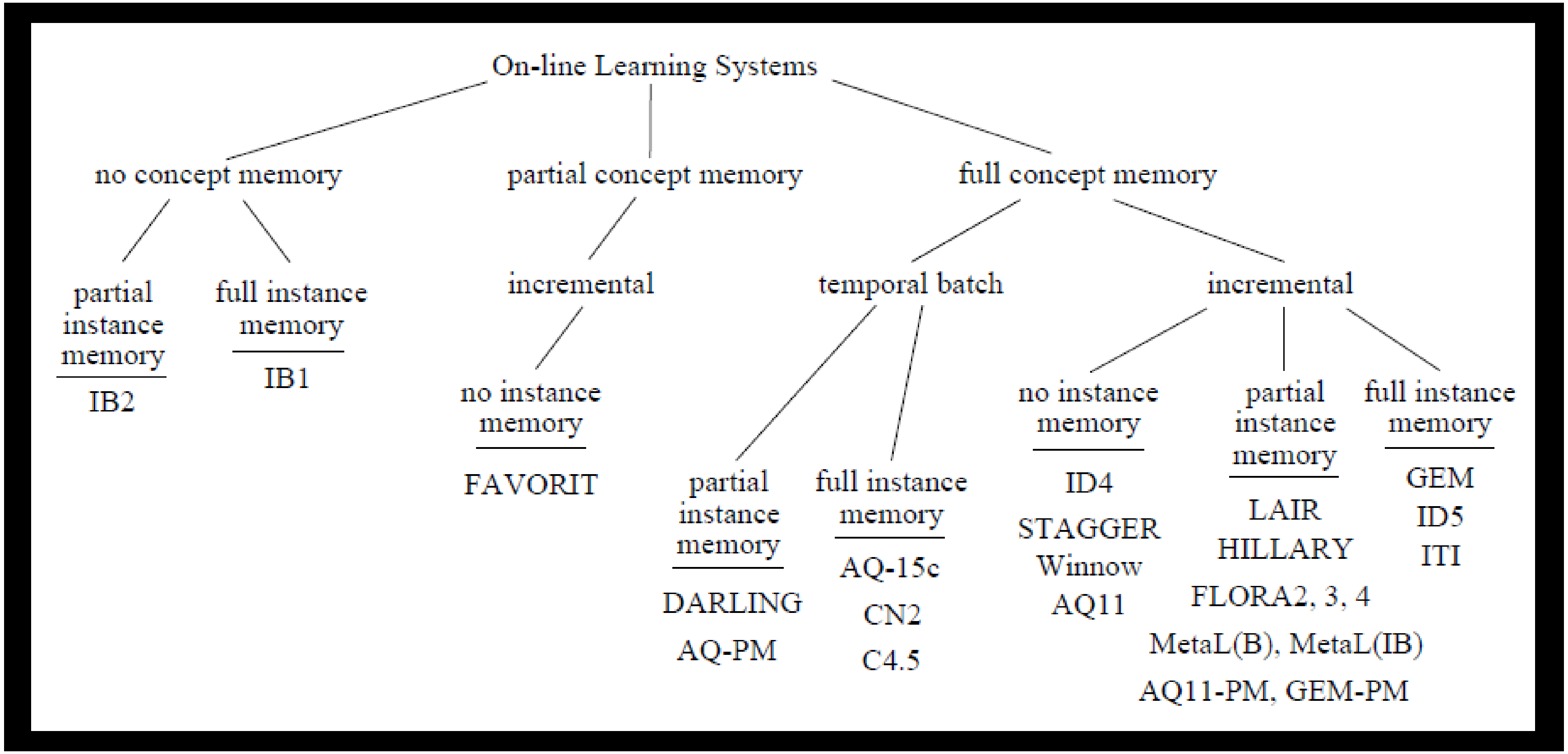
Few definitions

- Incremental learning
- Instance memory
- Concept memory
- Online learning
  - Incremental learning
  - Real time processing
  - Incoming order
  - Drift detection

Few definitions

- Incremental learning
- Concept memory
- Instance memory
- Online learning
- Concept drift
    - Hidden context changes

Representative approaches

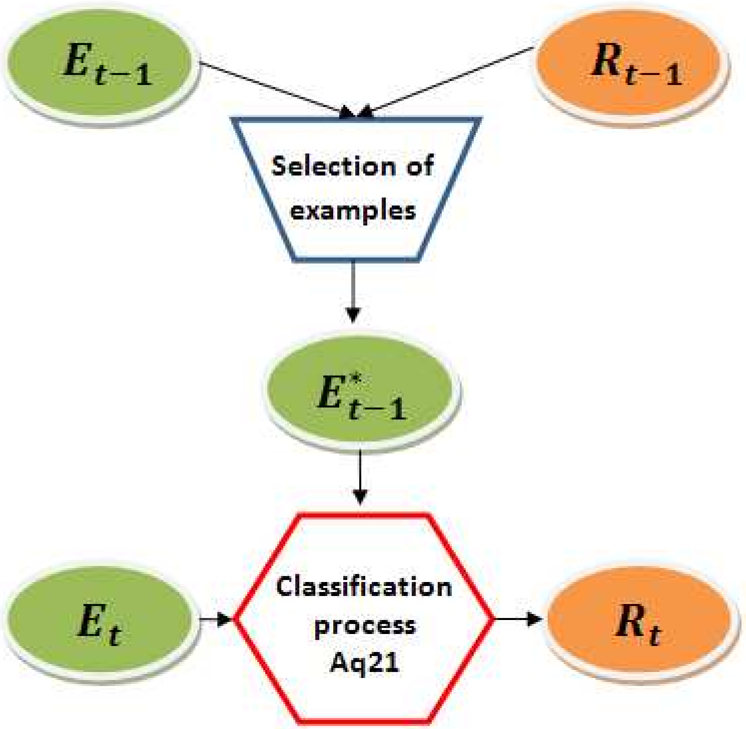


According to Maloof et all,2004[1]

A comparison between the representative incremental methods with partial instance memory

	AQ11-PM	FLORA	IB	FACIL	DARLING
Algorithm	Generalize training examples maximally	Generalize training examples when needed	Derived from nearest neighbor and do not generalize training examples	Derived from AQ11-PM	Make classification tree
Examples	Store only positive extreme examples	Store examples over a window of time	Store specific examples	Store both positive and negative examples, not necessarily extreme	Store specific neighboring examples
Interesting features		Keep old stable concepts	Use similarity function	Use the growth of a rule	Use a specific weight forgetting mechanism
Disadvantages	May cause overtraining	May delete available concept description	Computationally expensive	Has user defined parameters hard to tune	May not delete outdated concepts

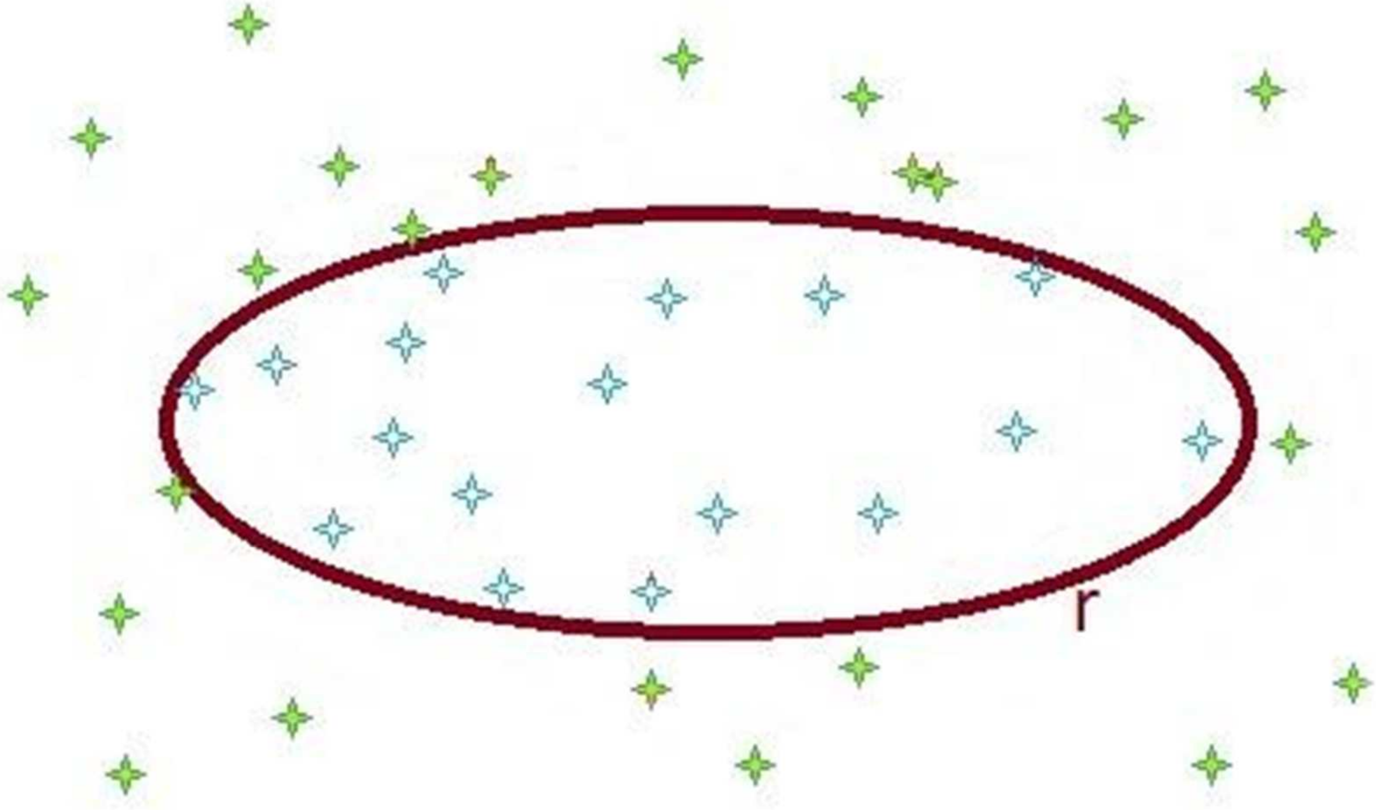
Proposed approach of incremental learning with partial instance memory and no concept memory



- **Classification basis non-incremental algorithm: AQ21**
- **Selection and storage of band border examples: Similar to AQ11-PM**

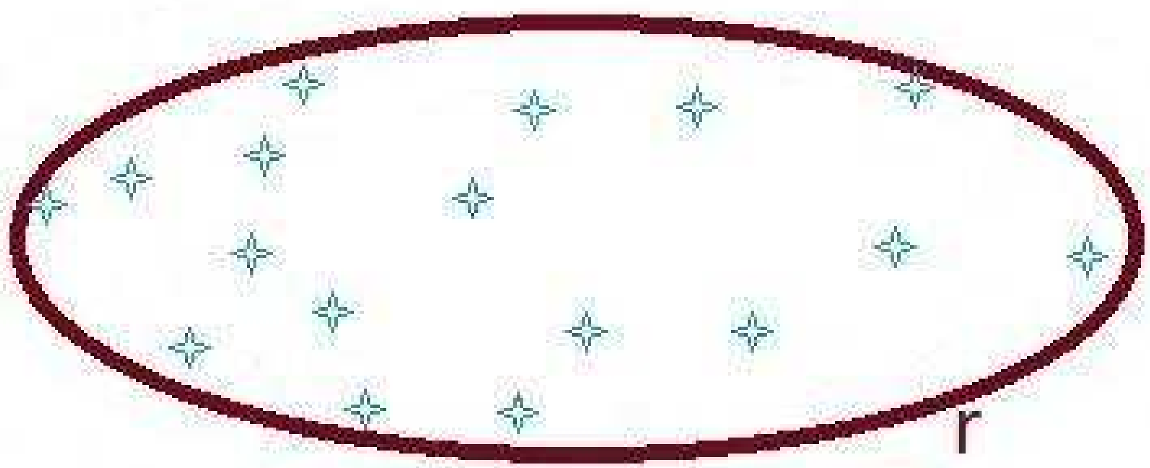
- Partial instance memory
  - the memory requirement is decreased and limited
  - the learning time is diminished
- Rule induction (AQ family)
  - the whole search space is not analyzed
  - the rules are quickly created and deleted
- Forgetting mechanism
  - the concept drifts are detected

Selection of examples



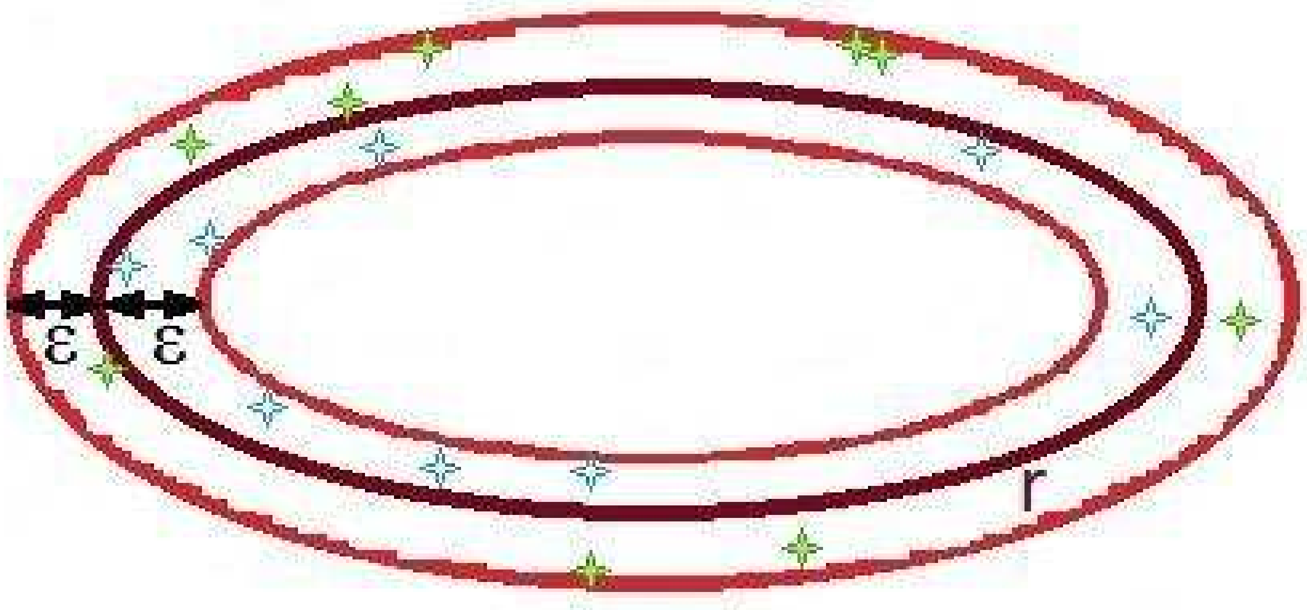
Selection of examples

- keep all examples of rules covering less than θ examples



Selection of examples

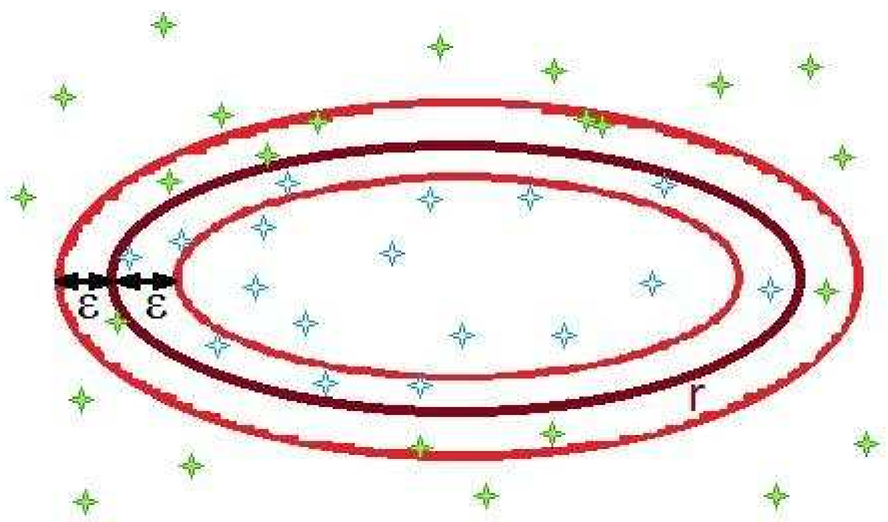
- keep band border examples, with the distance < ε, for rules covering over θ examples (similar to AQ11-PM)
- fix a maximum of stored examples (Ki for positive and Ke for negative)



- Distance function: $dist(r,e) = \sum_{i=1}^{m} d(c_i, x_i)$ $\qquad d(c_i, x_i) = \begin{cases} \min\{(c_{iu} - x_i), (c_{il} - x_i)\}\ normalized, if\ c_i \in N \\ \neg(x_i \in c_i), if\ c_i \in S \end{cases}$

Selection of examples

- keep all examples of rules covering less than θ examples
- keep band border examples, with the distance < ε, for rules covering over θ examples (similar to AQ11-PM)
- fix a maximum of stored examples (Ki for positive and Ke for negative)
- age forgetting mechanism (when needed)



- Distance function: $dist(r,e) = \sum_{i=1}^{m} d(c_i, x_i)$ $d(c_i, x_i) = \begin{cases} \min\{(c_{iu} - x_i),(c_{il} - x_i)\}\ normalized, if\ c_i \in N \\ \neg(x_i \in c_i), if\ c_i \in S \end{cases}$

What we want to assess by experimentations

- Non-incremental
  - Get results similar to behavior rules used for simulation
- Incremental in stationary environment
  - Keep the memory limited
  - Keep learning time almost constant
  - Get results similar to the non-incremental approach
- Incremental in a drifting environment
  - Achieve rules similar to behavior rules of current iteration
  - Detect and track drifts using a forgetting mechanism

Behavior rules for simulating input data

A total of 11 rules:

- General
  - If battery < 3% then crash low battery
  - If memory RAM > 95% then crash memory full
  - If memory ROM > ROM size – 1000 MB then crash memory full
- Operation system
  - If Appli_Incompatible_Android open and OS =Android then crash applicrash
  - If Appli_Incompatible_IOS open and OS =IOS then crash application
  - If Appli_Incompatible_MWP open and OS = MWP then crash application
- Brand
  - If brand=Sony and battery < 8% then crash low battery
  - If brand =Apple and Appli _GPS open and Appli_Incompatible_GPS open then crash application
- Model
  - If model = Omnia7 and Appli_Incompatible_Omnia open then crash application
  - If model =GalaxyMini and Appli _GSM open and Appli _WIFI open and Appli _GPS open and battery< 10% then crash low battery
- Specific
  - If Appli_Incompatible_Telephone open then crash application

Experimentations

❖ Non-incremental

Learning time	Total positive	Total negative	Total stored examples	Number of important rules	Total number of rules	Precision	Recall
60 m	11,211	145,757	156,968	12	24	100%	99.67%

❖ Incremental in a a stationary environment

- 6 steps of incremental learning
- input of one step include:
  - approximately 30.800 new incoming reports
    - » 88 different smartphones
  - approximately a 5 days simulation / smartphone
    - » 350 reports / smartphone

Empirical results

No	θ	ε	Ki	Ke	Time	Time last incremental step	Mean positive	Mean negative	Mean of stored examples	Number of important rules	Total number of rules	Precision	Recall
-	-	-	-	-	-	60 m	11,211	145,757	156,968	12	24	100%	99.67%
1	25	0	250	250	111 m 20 s	19 m 38 s	1,229.6	904.5	32,309.4	11	41	56.47%	93.22%
2	25	0.5	250	250	143 m 29 s	31 m 35 s	2,359.4	3,859.5	35,315.4	24	62	99.67%	95.07%
3	25	1	250	250	131 m 33 s	27 m 12 s	2341	5616	36,527.6	25	62	99.82%	96.75%
4	25	1.5	250	250	115 m 37 s	18 m 58 s	2,297.8	5,981.6	36,643.6	23	61	99.94%	96.47%
5	25	2	250	250	119 m 10 s	24 m 22 s	2,279.3	5,997.3	36,643.6	24	64	99.94%	96.47%
6	25	2.5	250	250	122 m 23 s	29 m 1 s	2,384.2	6,291.8	36,676.8	27	68	99.86%	96.56%
7	30	0	100	100	97 m 40 s	11 m 19 s	880.8	499.8	31,857.6	11	42	98.77%	93.66%
8	30	0.5	100	100	106 m 52 s	18 m 27 s	1,467.8	1,984.8	33,256.4	17	57	99.00%	96.29%
9	30	1	100	100	106 m 1 s	24 m 8 s	1,457.8	2,644.1	33,809.6	19	60	99.65%	95.29%
10	30	1.5	100	100	104 m 6 s	25 m 4 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
11	30	2	100	100	101 m 18 s	24 m 16 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
12	30	2.5	100	100	101 m 5 s	24 m 12 s	1,437.3	2,895.3	33,845.6	18	68	99.71%	93.28%
						23 m 11 s	1,855.3	3,753.3	34,782.2	20	57		

Empirical results

No	θ	ε	Ki	Ke	Time	Time last incremental step	Mean positive	Mean negative	Mean of stored examples	Number of important rules	Total number of rules	Precision	Recall
-	-	-	-	-	-	60 m	11,211	145,757	156,968	12	24	100%	99.67%
1	25	0	250	250	111 m 20 s	19 m 38 s	1,229.6	904.5	32,309.4	11	41	56.47%	93.22%
2	25	0.5	250	250	143 m 29 s	31 m 35 s	2,359.4	3,859.5	35,315.4	24	62	99.67%	95.07%
3	25	1	250	250	131 m 33 s	27 m 12 s	2341	5616	36,527.6	25	62	99.82%	96.75%
4	25	1.5	250	250	115 m 37 s	18 m 58 s	2,297.8	5,981.6	36,643.6	23	61	99.94%	96.47%
5	25	2	250	250	119 m 10 s	24 m 22 s	2,279.3	5,997.3	36,643.6	24	64	99.94%	96.47%
6	25	2.5	250	250	122 m 23 s	29 m 1 s	2,384.2	6,291.8	36,676.8	27	68	99.86%	96.56%
7	30	0	100	100	97 m 40 s	11 m 19 s	880.8	499.8	31,857.6	11	42	98.77%	93.66%
8	30	0.5	100	100	106 m 52 s	18 m 27 s	1,467.8	1,984.8	33,256.4	17	57	99.00%	96.29%
9	30	1	100	100	106 m 1 s	24 m 8 s	1,457.8	2,644.1	33,809.6	19	60	99.65%	95.29%
10	30	1.5	100	100	104 m 6 s	25 m 4 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
11	30	2	100	100	101 m 18 s	24 m 16 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
12	30	2.5	100	100	101 m 5 s	24 m 12 s	1,437.3	2,895.3	33,845.6	18	68	99.71%	93.28%
						23 m 11 s	1,855.3	3,753.3	34,782.2	20	57		

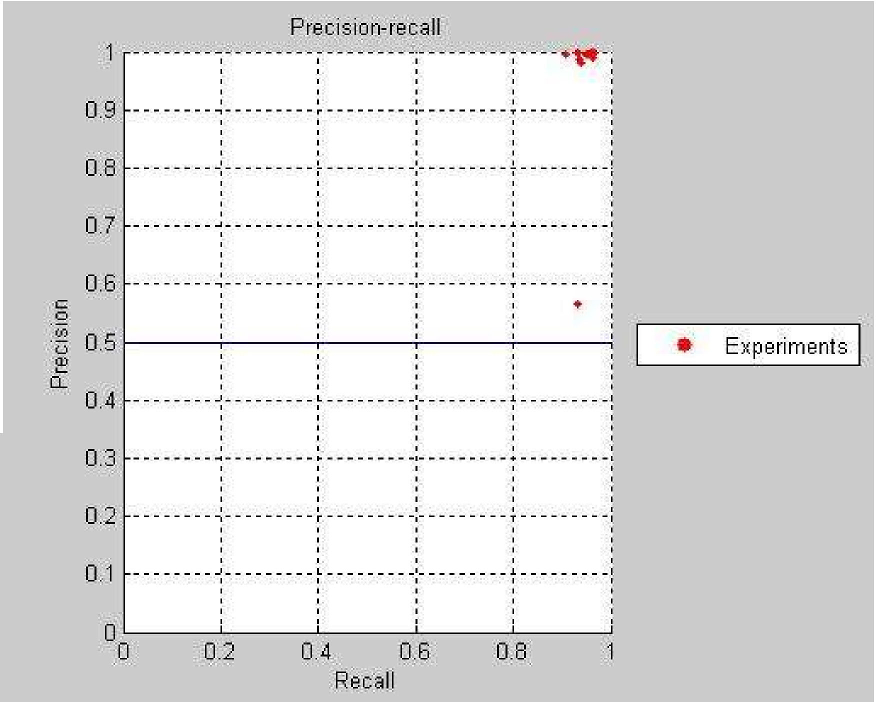
Empirical results

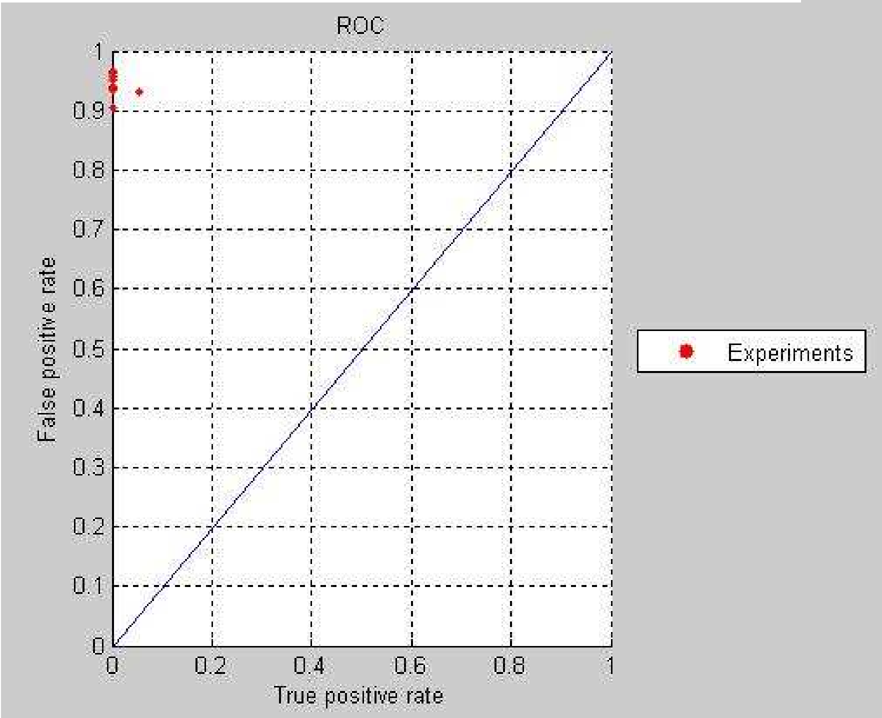
No	θ	ε	Ki	Ke	Time	Time last incremental step	Mean positive	Mean negative	Mean of stored examples	Number of important rules	Total number of rules	Precision	Recall
-	-	-	-	-	-	60 m	11,211	145,757	156,968	12	24	100%	99.67%
1	25	0	250	250	111 m 20 s	19 m 38 s	1,229.6	904.5	32,309.4	11	41	56.47%	93.22%
2	25	0.5	250	250	143 m 29 s	31 m 35 s	2,359.4	3,859.5	35,315.4	24	62	99.67%	95.07%
3	25	1	250	250	131 m 33 s	27 m 12 s	2341	5616	36,527.6	25	62	99.82%	96.75%
4	25	1.5	250	250	115 m 37 s	18 m 58 s	2,297.8	5,981.6	36,643.6	23	61	99.94%	96.47%
5	25	2	250	250	119 m 10 s	24 m 22 s	2,279.3	5,997.3	36,643.6	24	64	99.94%	96.47%
6	25	2.5	250	250	122 m 23 s	29 m 1 s	2,384.2	6,291.8	36,676.8	27	68	99.86%	96.56%
7	30	0	100	100	97 m 40 s	11 m 19 s	880.8	499.8	31,857.6	11	42	98.77%	93.66%
8	30	0.5	100	100	106 m 52 s	18 m 27 s	1,467.8	1,984.8	33,256.4	17	57	99.00%	96.29%
9	30	1	100	100	106 m 1 s	24 m 8 s	1,457.8	2,644.1	33,809.6	19	60	99.65%	95.29%
10	30	1.5	100	100	104 m 6 s	25 m 4 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
11	30	2	100	100	101 m 18 s	24 m 16 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
12	30	2.5	100	100	101 m 5 s	24 m 12 s	1,437.3	2,895.3	33,845.6	18	68	99.71%	93.28%
						23 m 11 s	1,855.3	3,753.3	34,782.2	20	57		

Empirical results

No	θ	ε	Ki	Ke	Time	Time last incremental step	Mean positive	Mean negative	Mean of stored examples	Number of important rules	Total number of rules	Precision	Recall
-	-	-	-	-	-	60 m	11,211	145,757	156,968	12	24	100%	99.67%
1	25	0	250	250	111 m 20 s	19 m 38 s	1,229.6	904.5	32,309.4	11	41	56.47%	93.22%
2	25	0.5	250	250	143 m 29 s	31 m 35 s	2,359.4	3,859.5	35,315.4	24	62	99.67%	95.07%
3	25	1	250	250	131 m 33 s	27 m 12 s	2341	5616	36,527.6	25	62	99.82%	96.75%
4	25	1.5	250	250	115 m 37 s	18 m 58 s	2,297.8	5,981.6	36,643.6	23	61	99.94%	96.47%
5	25	2	250	250	119 m 10 s	24 m 22 s	2,279.3	5,997.3	36,643.6	24	64	99.94%	96.47%
6	25	2.5	250	250	122 m 23 s	29 m 1 s	2,384.2	6,291.8	36,676.8	27	68	99.86%	96.56%
7	30	0	100	100	97 m 40 s	11 m 19 s	880.8	499.8	31,857.6	11	42	98.77%	93.66%
8	30	0.5	100	100	106 m 52 s	18 m 27 s	1,467.8	1,984.8	33,256.4	17	57	99.00%	96.29%
9	30	1	100	100	106 m 1 s	24 m 8 s	1,457.8	2,644.1	33,809.6	19	60	99.65%	95.29%
10	30	1.5	100	100	104 m 6 s	25 m 4 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
11	30	2	100	100	101 m 18 s	24 m 16 s	1,437.3	2,895.5	33,845.6	18	68	99.71%	93.28%
12	30	2.5	100	100	101 m 5 s	24 m 12 s	1,437.3	2,895.3	33,845.6	18	68	99.71%	93.28%
						23 m 11 s	1,855.3	3,753.3	34,782.2	20	57		

Empirical results





The overtraining impact

- The rules achieved incrementally have a good precision-recall, but are not as general as those retrieved in non-incremental, because of border examples stored
  - The non-incremental rule:
    - (batterie<= 7) and (brand = 'Sony' or brand = 'Nokia') and (numero<= 50000000)
  - The incremental set of 12 rules:
    - (batterie between 4 and 7) and (brand = 'Sony') and ("app_Appli_GPS" = 'running')
    - (batterie <= 7) and (brand = 'Sony') and (memoirephysique >= 677) and ("app_Appli_WIFI" = 'running')
    - (batterie <= 7) and (brand = 'Sony' or brand = 'Apple') and (memoirephysique <= 1400) and ("app_Appli_Call" = 'running')
    - (batterie between 4 and 7) and (brand = 'Sony' or brand = 'Apple') and (memoirephysique >= 1402) and ("app_Appli_Call" = 'running')
    - (batterie <= 7) and (brand = 'Sony' or brand = 'Apple') and (memoirephysique between 1402 and 1872) and ("app_Appli_Call" = 'running')
    - …

Experimentations in a drifting environment

- ❖ The fleet of smarphones are replaced each time we recomputed concept description and we pass from a smartphone model to another at each incremental step
- ❖ The age forgetting mechanism is used
  - 8 steps of incremental learning
  - input of one step include:
    - approximately 20.000 reports
    - all reports generated for one smartphone model
      - » **Models:** GalaxyMini, Galaxy S2, IPhone 4S, Omnia 7,
        Lumia 900, Lumia 800, Xperia Pro, Xperia Mini
    - each model includes 11 different smartphones simulations, during a month
- o In experiments the age parameter of the forgetting mechanism is set to 3

Empirical results

Experiment 1: Lumia 800

Models order: GalaxyMini, Galaxy S2, Xperia Pro, Xperia Mini, IPhone 4S, Omnia 7, Lumia 900, Lumia 800

- Behavior rules for the last iteration
  - (batterie < 3)
  - (memory RAM > 950)
  - (memory ROM > 15000)
  - ("app_Appli_Incompatible_MicrosoftWindowsPhone" = 'running')
  - ("app_Appli_Incompatible_Telephone" = 'running')

- Rules achieved
  - (batterie <= 2)
  - (memoirevive >= 922)
  - ("app_Appli_Incompatible_MicrosoftWindowsPhone" = 'running')
  - ("app_Appli_Incompatible_Telephone" = 'running')

Empirical results

Experiment 2: Xperia Mini

Models order: GalaxyMini, Galaxy S2, IPhone 4S, Omnia 7,
Lumia 900, Lumia 800, Xperia Pro, Xperia Mini

- Behavior rules for the last iteration
  - (batterie < 3)
  - (memory RAM > 950)
  - (memory ROM > 3000)
  - (" app_Appli_Incompatible_Android " = 'running')
  - (batterie < 8) and (modelu = XperiaMini')
  - ("app_Appli_Incompatible_Telephone" = 'running')

- Rules achieved
  - (batterie <= 7) and (modelu <> 'Lumia800')
  - (memoirevive >= 933)
  - (batterie <= 2)
  - (memoirevive <= 543) and ("app_Appli_Incompatible_Android" = 'running')
  - (version <> 'Android23') and ("app_Appli_Incompatible_MicrosoftWindowsPhone" = 'running')
  - ("app_Appli_Incompatible_Telephone" = 'running')

Conclusions

❖ The proposed approach

- use simulated data
- limit the memory requirements
- keep learning time almost constant
- store band border examples (similar to AQ11-PM)
- incorporate a forgetting mechanism
- detect and adapt to concept drifts

Future work

- Improve feature selection
  - » drop irrelevant dimensions
- Develop a concept memory approach
  - » filter rules according to some specific parameters
  - » limit the number of saved rules for each iteration
  - » merge current rules with previous ones
- Deal with examples which are masked by prevention rules

Thank you !