Data Streaming for Autonomic Computing in the EGEE framework

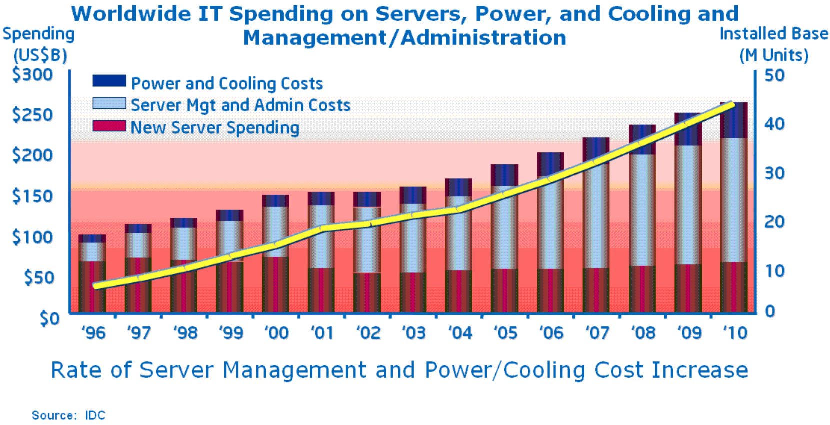
Xiangliang Zhang, Cyril Furtlehner, Michèle Sebag

TAO – INRIA CNRS
Université de Paris-Sud, F-91405 Orsay Cedex, France

Contents

Contents

Motivations of Autonomic Computing

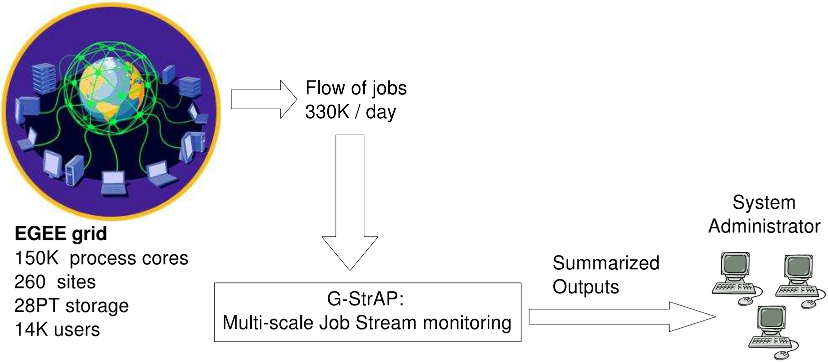


AUTONOMIC VISION & MANIFESTO
http://www.research.ibm.com/autonomic/manifesto/

Self-managing system with the ability of

- Self-healing: detect, diagnose and repair problems
- Self-configuring: automatically incorporate and configure components
- Self-optimizing: ensure the optimal functioning wrt defined requirements
- Self-protecting: anticipate and defend against security breaches

Autonomic Grid Computing System



EGEE: Enabling Grids for E-sciencE, http://www.eu-egee.org
EGEE User Forum: annual event since 2007

Job stream monitoring by clustering

Goal: **summarizing** the **large scale** and **fast arriving** data.

- provide **compact description**
- help to find out **interesting patterns**
- **classify** the incoming data

Challenges:

- **Large size**
  - **save** all the data and process them **as a whole** ?
    require **huge** disk, CPU, and memory (impossible for data in size of GB, TB, even PB, ..)
  - process the data **part by part** ?
    how to guarantee the **global optimization**.
- **Changing distribution**:
  for the time-ordered data, how to make the clusters **keep tracking the evolving data**?

What is Clustering ?

- unsupervised learning method
- group similar points together in the same group (cluster)
- widely used on various problems:
  Interesting groups discovery, Data structure presentation, Data classification, Data compression, Dimensionality reduction or feature selection
- many clustering methods are available, e.g., Hierarchical clustering methods, Density-based methods(Dbscan), Partitioning methods(k-means)

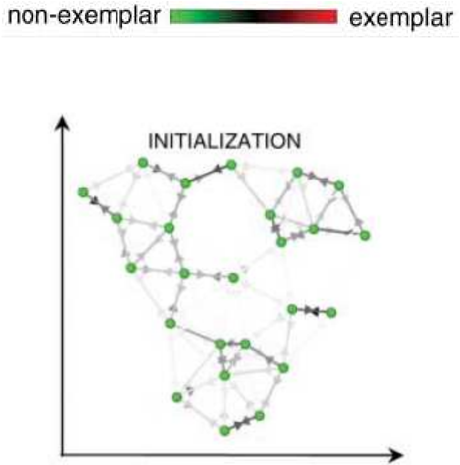
Our requirements of clustering method

- No need to set the number K of clusters *double-edged sword*
- global optimization of clustering result:
  not locally optimized by greedy approach
- stable clustering result:
  not affected by the initialization
- real data points as **representative exemplars** (cluster center):
  suit the application field when averaged centers are meaningless,
  e.g. molecule, jobs described by categorical attributes

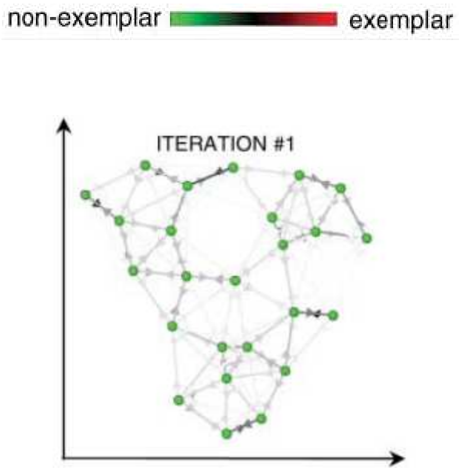
Our requirements of clustering method

- No need to set the number K of clusters *double-edged sword*
- global optimization of clustering result:
  not locally optimized by greedy approach
- stable clustering result:
  not affected by the initialization
- real data points as **representative exemplars** (cluster center):
  suit the application field when averaged centers are meaningless,
  e.g. molecule, jobs described by categorical attributes
- **Affinity Propagation (AP)** (Frey & Dueck, Science2007)

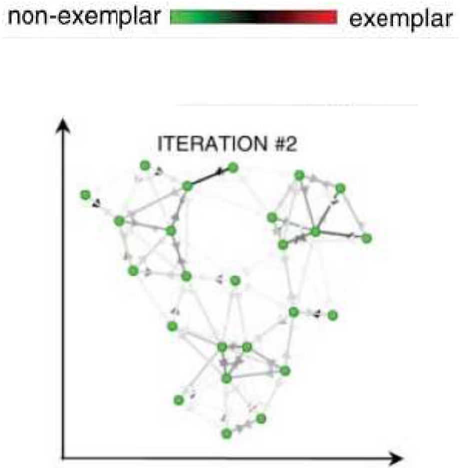
Iterations of Message passing in AP



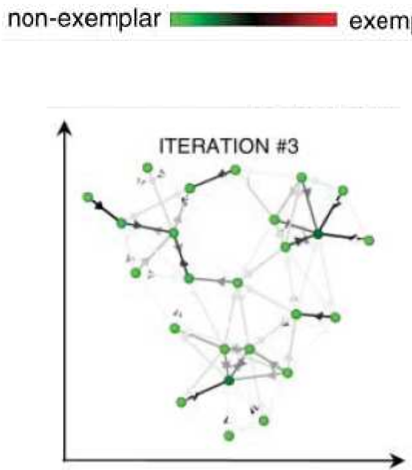
Iterations of Message passing in AP



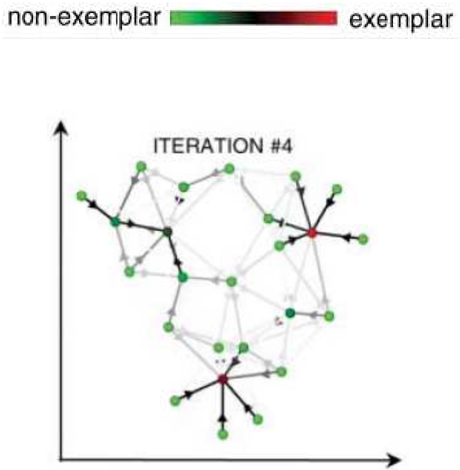
Iterations of Message passing in AP



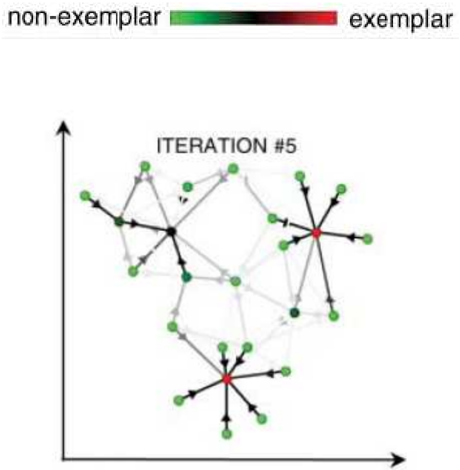
Iterations of Message passing in AP



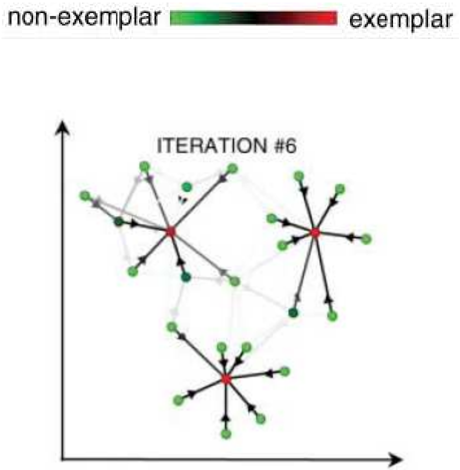
Iterations of Message passing in AP



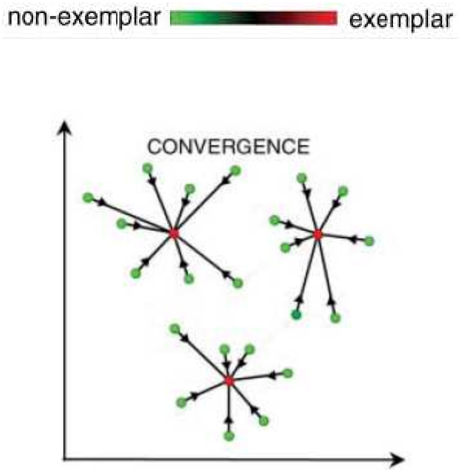
Iterations of Message passing in AP



Iterations of Message passing in AP



Iterations of Message passing in AP



Introduction of AP

input:

Data: $x_1, x_2, ..., x_N$ Distance: $d(x_i, x_j)$

find:

σ: $x_i \longrightarrow \sigma(x_i)$, exemplar representing x_i, such that

$$max \sum_{i=1}^{N} S(x_i, \sigma(x_i))$$

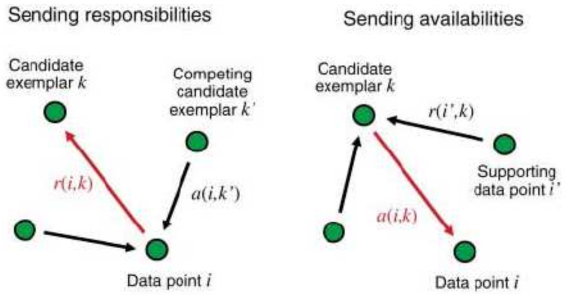
where,

$S(x_i, x_j) = -d^2(x_i, x_j)$ if $i \neq j$

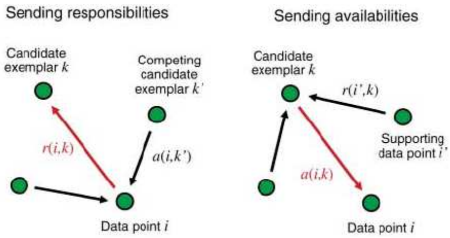
$S(x_i, x_i) = -s^*$ s^*: user-defined parameter (penalty)

- $s^* = \infty$, only one an exemplar (one cluster)
- $s^* = 0$, every point is an exemplar (N clusters)

AP: a message passing algorithm



Message passed



$r(i,k) = S(x_i, x_k) - \max_{k', k' \neq k}\{a(i,k') + S(x_i, x'_k)\}$

$r(k,k) = S(x_k, x_k) - \max_{k', k' \neq k}\{S(x_k, x'_k)\}$

$a(i,k) = min\left\{0, r(k,k) + \sum_{i', i' \neq i,k} \max\{0, r(i',k)\}\right\}$

$a(k,k) = \sum_{i', i' \neq k} \max\{0, r(i',k)\}$

The index of exemplar $\sigma(x_i)$ associated to x_i is finally defined as:

$\sigma(x_i) = argmax\left\{r(i,k) + a(i,k), k = 1 \dots N\right\}$

Summary of AP

Affinity Propagation (AP)

- A clustering method
- Converge by Iterations of Message passing
- No need of K (the number of clusters)
- Real point as exemplar
- an application of *belief propagation* (simplified graph + message passing)

cons

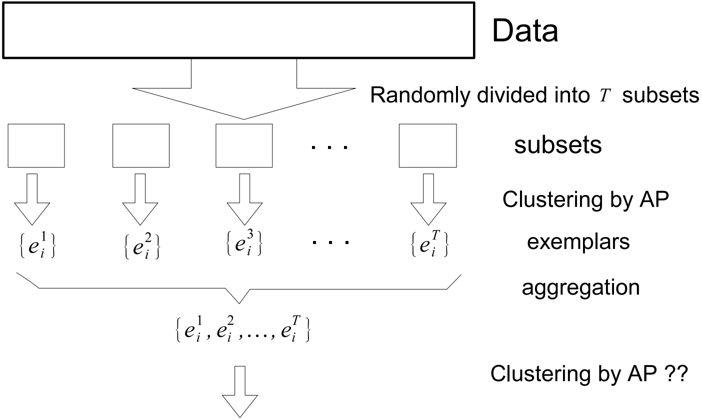
Computational complexity problems

- Similarity computation: $\mathcal{O}(N^2)$
- Message passing: $\mathcal{O}(N^2 \log N)$

Contents

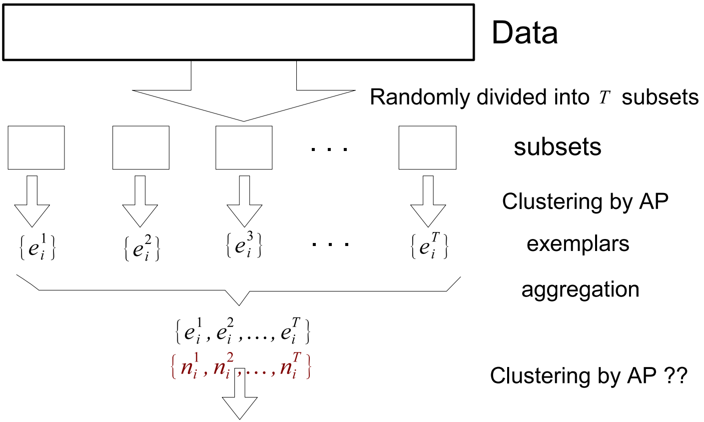
Hierarchical AP

Divide-and-conquer (inspired by Guha et al, TKDE2003)



Hierarchical AP

Divide-and-conquer (inspired by Guha et al, TKDE2003)



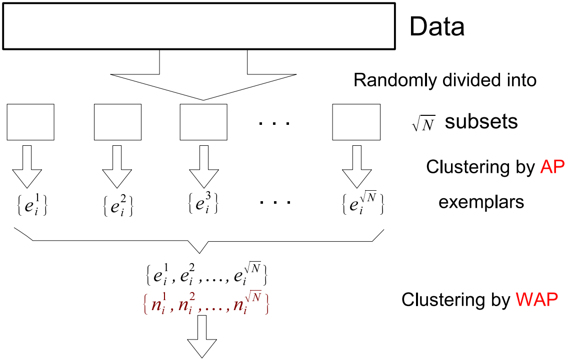
Weighted AP

AP		WAP
x_i		x_i, n_i
$S(x_i, x_j)$	$\longrightarrow$	$n_i \times S(x_i, x_j)$
price for x_i to select x_j as an exemplar		
$S(x_i, x_i)$	$\longrightarrow$	$S(x_i, x_i) + (n_i - 1) \times \epsilon$
price to select x_i as exemplar		ϵ is variance of n_i points

Proposition

WAP $\equiv$ AP with duplications (aggregations)

Hierarchical AP



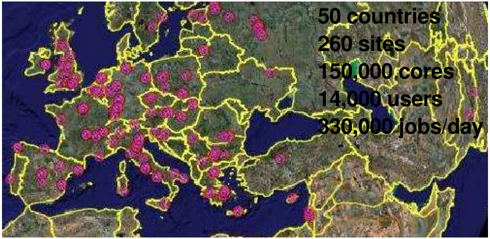
- Complexity of HI-AP is $\mathcal{O}(N^{3/2})$

  (X. Zhang et al, ECML/PKDD 2008)
- NB: can be iteratively reduced to $\mathcal{O}(N^{1+\gamma})$

  (X. Zhang et al, SIGKDD 2009)

Validation of Hi-AP on EGEE jobs

- EGEE (Enabling Grids for E-sciencE)
- Grid Observatory http://www.grid-observatory.org/



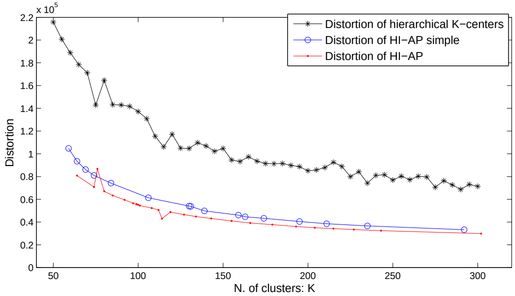
description of jobs (237,087)

- 4 numeric features: duration of execution
- 1 symbolic feature: name of queue

Validation of HI-AP on EGEE jobs

Evaluation: Distortion

$$D([\sigma]) = \sum_{i=1}^{N} d^2(x_i, \sigma(x_i))$$



- 237,087 jobs
- 10 mins on Intel 2.66GHz Dual-Core PC with 2 GB memory

HI-AP has the lowest distortion compared to baseline method

Contents

Challenges of Stream Clustering

Data stream:

a **real-time**, **continuous**, **ordered** sequence of items arriving at a very **high speed** (Golab & Özsu,SigMod2003)

e.g., network traffic data, sensor network monitoring data

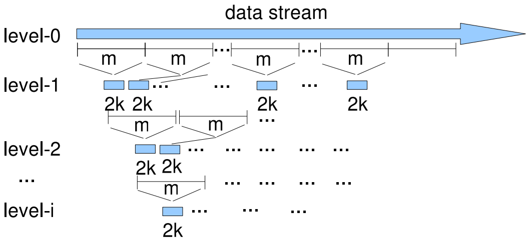
Data streams clustering

- Provide **compact** description of data flow
- **Incremental** model updating
- **No** specified **number of clusters**
- Process in **real-time**
- **Available** results at **any time**

Related works

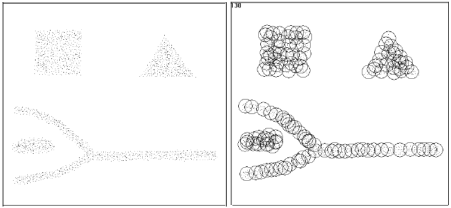
Divide-and-conquer strategy (Guha et al, TKDE 2003)

fixed segmentation window ——— > not feasible to handle the changing distribution



Related works

A two-level scheme (Aggarwal et al, VLDB 2003)



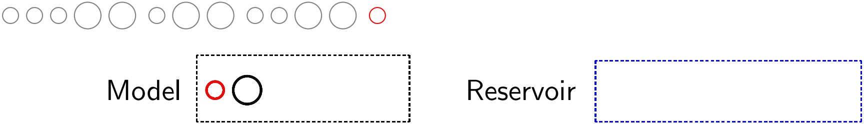
- **online level** to summarize the evolving data stream
- **offline level** to generate the clusters using the summary.
- **clustering** method is used to get **initial** micro-clusters and **final** clusters. e.g., Density-based clustering methods DBSCAN (Cao et al, SDM 2006)

Problem: the online clustering models is not provided or only available when it is required by users.

Contents

Stream clustering

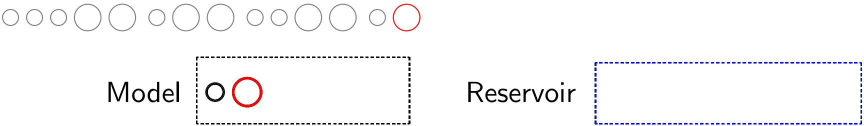
Stream clustering



Does x_t fit the current model ??

- if yes, update the model
- otherwise, go to reservoir

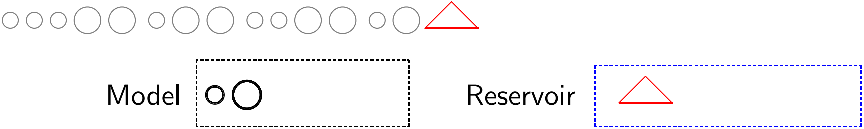
Stream clustering



Does x_t fit the current model ??

- if yes, update the model
- otherwise, go to reservoir

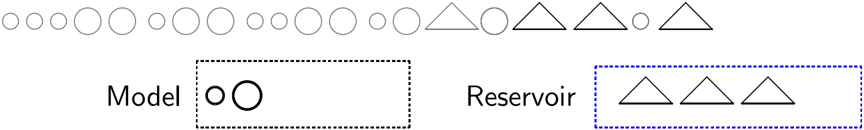
Stream clustering



Does x_t fit the current model ??

- if yes, update the model
- otherwise, go to reservoir

Stream clustering

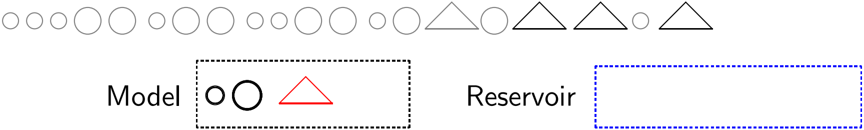


Has the distribution changed ??

CHANGE TEST

- if yes, rebuild the model
- otherwise, continue

Stream clustering



Has the distribution changed ??

CHANGE TEST

- if yes, rebuild the model
- otherwise, continue

StrAP Method

data → data streaming process system → models $\{\ e_i,\ n_i, \Sigma_i,\ t_i\ \}$

Does x_t fit the current model ??

- if yes, update the model   update the weight with time decay (decay window Δ)
- otherwise, go to reservoir

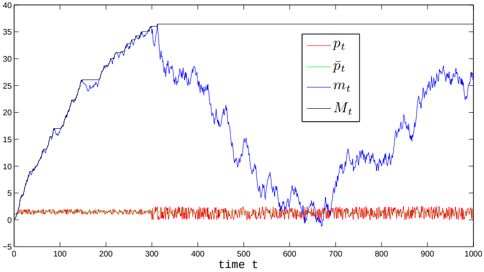
Has the distribution changed ??

- if yes, rebuilt the model   based on current model and reservoir by WAP
- otherwise, continue

Rebuild the model??

- when reservoir is full
- when changes are detected: Page-Hinkley statistic (Cumulative-Sum-like test)

(Page, Biometrika1954; Hinkley, Biometrika1971)



p_t changing distribution

$$\bar{p}_t = \frac{1}{t}\sum_{\ell=1}^{t} p_\ell$$

$$m_t = \sum_{\ell=1}^{t}(p_\ell - \bar{p}_\ell + \delta)$$

$$M_t = max\{m_\ell\}$$

$$PH_t = M_t - m_t$$

if $PH_t > \lambda$, changed detected

How to set λ ???

Setting of λ

- fixed empirical value (X. Zhang et al, ECML/PKDD 2008)
- self-adaptive change detection test (X. Zhang et al, SIGKDD 2009)

Self-adapt λ $\equiv$ An optimization problem

BIC: $\mathcal{F}_\lambda = \frac{1}{|C|}\sum_{i=1}^{|C|}\left(\frac{1}{n_i}\sum_{e_j\in C_i} d(e_j, e_i^*)\right) + \varphi\frac{\rho}{2}\log N + \eta O_t$
$\propto$ loss $+$ size of model $+$ percentage of outlier

OPTIMIZATION:

- ϵ-greedy search from a finite set of λ values

$$\lambda = argmin\{\mathbf{E}(F_\lambda)\},$$

λ_1	λ_2	λ_3	λ_4	...
$\mathbf{E}(F_{\lambda_1})$	$\mathbf{E}(F_{\lambda_2})$	$\mathbf{E}(F_{\lambda_3})$	$\mathbf{E}(F_{\lambda_4})$	...

- Gaussian Process Regression based on $\{\lambda_i, F_{\lambda_i}\}$
  continuous value of λ is generated

Contents

Validation of StrAP on KDD99 data

Data used

- Real world data: KDD99 data
  - intrusion detection benchmark
  - 494,021 network connection records in $\mathbb{R}^{34}$
  - 23 classes: 1 normal + 22 attacks
- Baseline: *DenStream* (Cao et al, SDM2006)

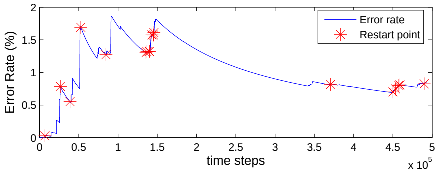
Performance indicator (supervised setting)

- Clustering accuracy
- Clustering purity

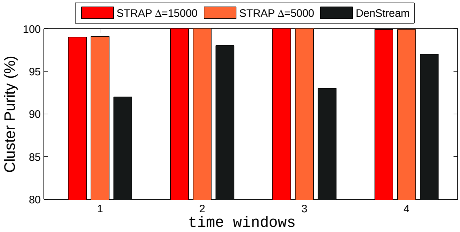
KDD Cup 1999 data: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Accuracy and Purity along time

Error Rate along time $< 2\%$



Higher clustering purity than DenStream



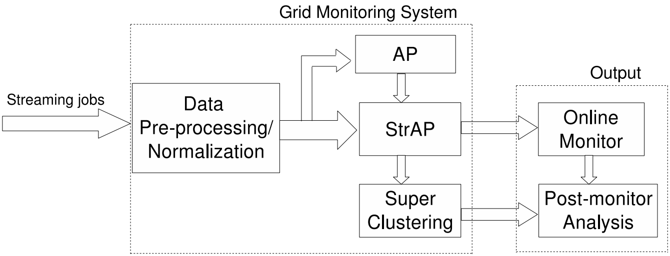
Discussion

StrAP vs *DenStream*

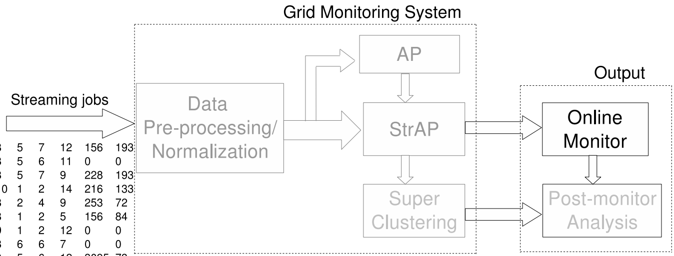
- Pros
  - better accuracy
    Truth Detection rate: 99.18%
    False Alarm rate: 1.39%
    Online Error rate $< 2\%$
  - model available at any time
- Cons
  - *DenStream*: 7 seconds
  - StrAP : 7 mins

Contents

Multi-scale Realtime Grid Monitoring System

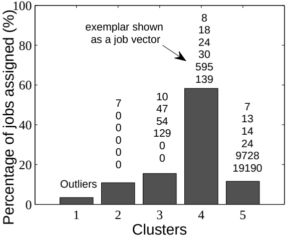


Multi-scale Realtime Grid Monitoring System

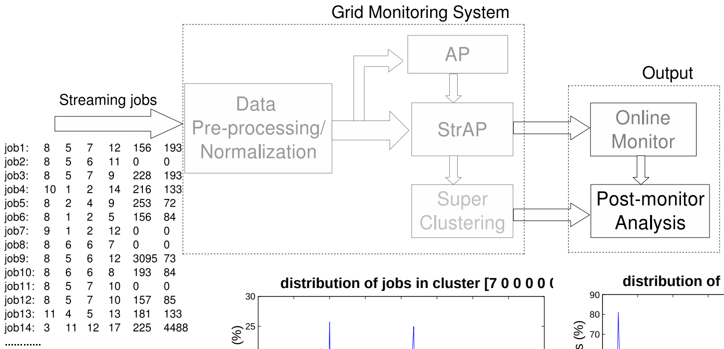


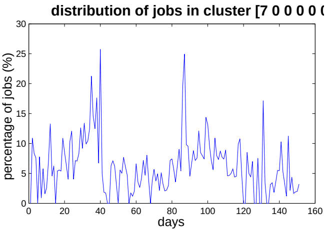
job1:	8	5	7	12	156	193
job2:	8	5	6	11	0	0
job3:	8	5	7	9	228	193
job4:	10	1	2	14	216	133
job5:	8	2	4	9	253	72
job6:	8	1	2	5	156	84
job7:	9	1	2	12	0	0
job8:	8	6	6	7	0	0
job9:	8	5	6	12	3095	73
job10:	8	6	6	8	193	84
job11:	8	5	7	10	0	0
job12:	8	5	7	10	157	85
job13:	11	4	5	13	181	133
job14:	3	11	12	17	225	4488

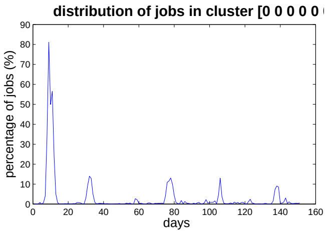
....



Multi-scale Realtime Grid Monitoring System







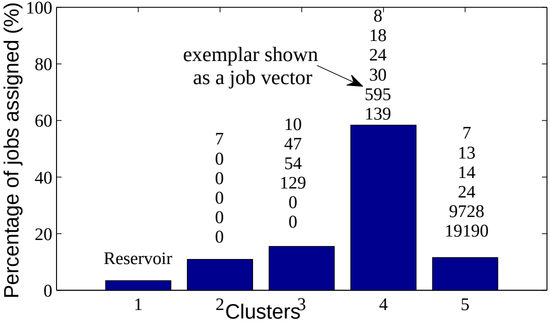
Experimental Data

- EGEE logs of 39 RBs during 5 months (2006-01-01 ∼ 2006-05-31)
- 5,268,564 jobs
- for each job, its
  - final status (good or type of errors)
  - 6 features describing the time-cost of services in a job lifecycle

Experimental Results: Online Monitoring outputs

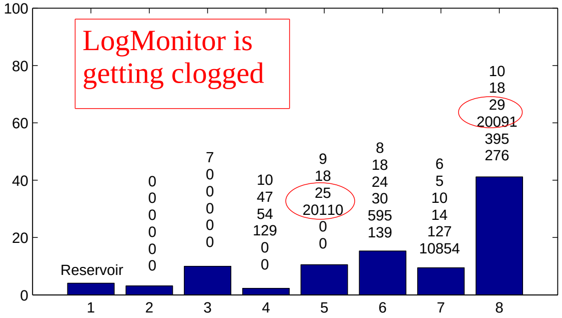
Real-time Monitoring: when change detected

Online summarizing the streaming jobs into clusters:

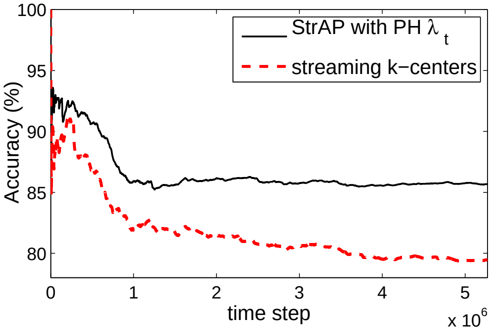


Real-time Monitoring: when change detected

Online summarizing the streaming jobs into clusters:



Clustering Accuracy



10% higher than baseline method(Streaming *k*-centers)

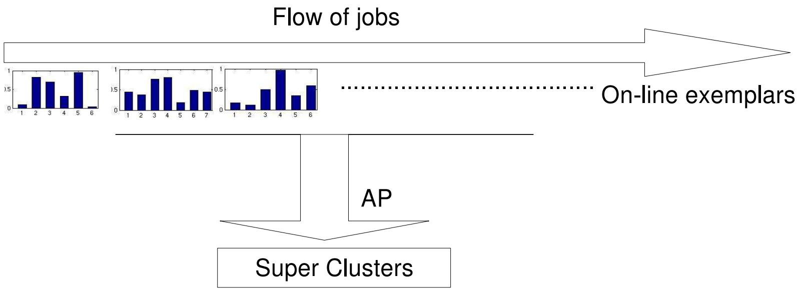
Discussion

- Real-time quality (330K jobs/day):
  - tested on Intel 2.66GHz Dual-Core PC with 2 GB memory
  - **10k jobs per minute** coding in **Matlab**
  - **60k jobs per minute** coding in **C/C++**
- concise online summary of the streaming jobs, with
  - proportion of defects
  - performance of the grid services

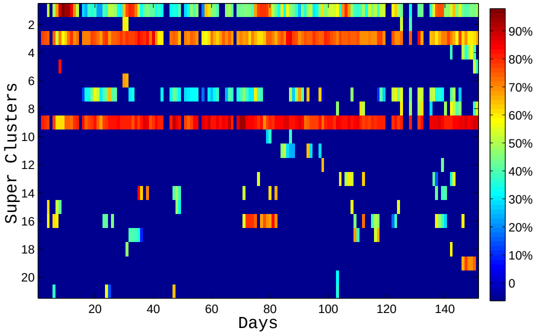
Experimental Results: Offline Analysis

Large-time scale Monitoring: Global view

- the history behavior of interesting exemplars
- without prior knowledge about failure patterns
- summarizing Gbyte data



Bad Super Exemplars: day view



"early stopped error", **Who and When** ?

Date	Jan 7∼13	Jan 30 ∼ Feb 3	Mar 16∼21	May 17∼19
UserID	A1	A1	B1	D1 and A1

Discussion and Conclusion

- real-time monitoring Grid job streams
- providing multi-scale models to describing the status of Grid
  - proportion of different type of job patterns (realtime-view, day-view, week-view ....)
  - rupture steps
  - offline globally analysis
- good quality clustering is guaranteed

Contents

Conclusion, Algorithm

Scalability: Hi-AP

- Reduce complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N^{3/2})$
- Iteratively reduce toward $\mathcal{O}(N^{(1+\gamma)})$

Stream clustering: StrAP

- Framework of processing the streaming data
- Hybridized with an efficient change detection method, Page-Hinkley
- Model available at any time
- BUT: slower than DenStream

Conclusion, Application

Network Intrusion Detection (KDD99 data)

- clustering by **one-scan** of the data
- using only $< 1\%$ data for building model **Active Learning**
- **high** clustering and classification **accuracy**

Autonomic Grid Computing

- real-time grid monitoring system
- **visualized online output** describing grid running status
- **offline output** for historical performance analysis
- multi-scale analysis of system behaviors

Ongoing work

Flexible Clustering Methods

- Fixed number clusters by messaging passing
- Arbitrary shape clusters by messaging passing
- Comprehensive model of streaming data
  using several representative exemplars covering the cluster, instead of one center point

Online Learning

- Assess the alarm level attached to a given model
  criticality of the clusters based on its frequency along time
- User profiling
  the clusters —> new features —> describe the users (viewing a user as a set of clusters)

Thank you for your attention.

Xiangliang ZHANG

xlzhang@lri.fr

http://www.lri.fr/∼xlzhang