

# Contributions to Large Scale Data Clustering and Streaming with Affinity Propagation. Application to Autonomic Grids.

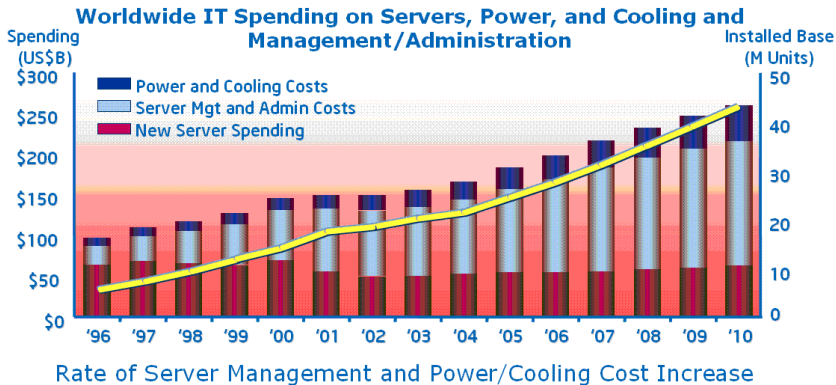
Xiangliang Zhang

Direction de thèse : Michèle Sebag et Cecile Germain-Renaud

TAO – LRI, INRIA, CNRS  
Université de Paris-Sud

July 28, 2010

# Motivations: Autonomic Computing



Source: IDC

Major part of the cost: management

## **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 w.r.t defined requirements
- ▶ **Self-protecting**: anticipate and defend against security breaches

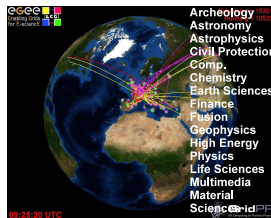
How:

- ▶ pre-requisite is to have a model of the system behavior
- ▶ there is no model based on first principles

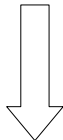
Machine Learning and Data Mining for Autonomic Computing

[Rish et al., 2005]

# Autonomic Grid Computing System



Flow of jobs  
330K / day



**EGEE grid**  
150K process cores  
260 sites  
28PT storage  
14K users

**G-StrAP:**  
Multi-scale Job Stream monitoring

Summarized  
Outputs

System  
Administrator



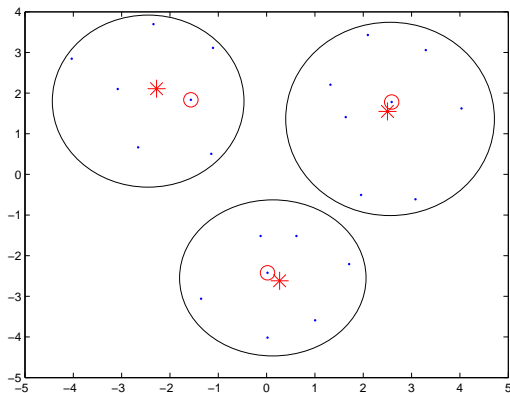
EGEE: Enabling Grids for E-science, <http://www.eu-gee.org>  
Infrastructure project, DataGrid(2002-2004), EGEE-I(2004-2006),  
EGEE-II(2006-2008), EGEE-III(2008-2010) and EGI(2010-2013)

# Summarizing a dataset

- ▶ **Clustering** : grouping similar points in the same group (cluster)
- ▶ Extracting **Exemplars**: real objects from dataset

better suited to complex application domains (e.g., molecules, structured items)

\* is the averaged center; ○ is the exemplar



# Position of the problem

Given:

**Data:**  $\mathcal{E} = \{x_1, x_2, \dots, x_N\}$       Distance:  $d(x_i, x_j)$

Define:

**Exemplars:**  $\{e_i\}$  is a subset of  $\mathcal{E}$

**Distortion:**

$$D(\{e_i\}) = \sum_{i=1}^N \min_{e_i} (d^2(x_i, e_i))$$

Goal:

Find a **mapping**  $\sigma, x_i \rightarrow \sigma(x_i) \in \{e_i\}$   
minimizing the distortion

**NB:**

Combinatorial optimization problem (NP).

**Job stream:** jobs submitted by the grid users at  $24 * 7$ ,  
more than 200 jobs/min

How to make a summary of the job stream ?

## Features

streaming of jobs

arriving fast

user-visible

non-stationary distribution

## Requirements

actual jobs as exemplars for traceability

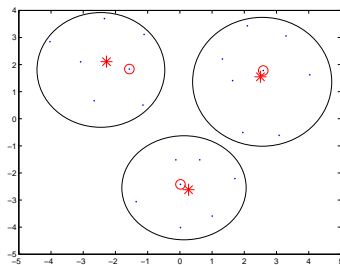
real-time processing

model available at any time

change detection

- ▶ Motivations
- ▶ **Clustering:**
  - The State of the Art
  - Large-scale Data Clustering
- ▶ **Streaming:** Data streams Clustering
- ▶ **Application** to Autonomic Computing:
  - A Multi-scale Real-time Grid Monitoring System
- ▶ Conclusions and Perspectives

# Clustering: The State of the Art



- ▶ **Averaged centers:** [Bradley et al., 1997]
  - $k$ -means, minimizing the **sum-squared distance** from a point to its center
  - $k$ -medians, minimizing the **sum of distance** from a point to its center
  - $k$ -centers, minimizing the **maximum distance** from a point to its center
- ▶ **Exemplars:** [Kaufman and Rousseeuw, 1987]
  - minimizing the **sum-squared distance** from a point to its exemplar
  - $k$ -medoids, [Kaufman and Rousseeuw, 1990, Ng and Han, 1994]
  - Affinity Propagation [Frey and Dueck, 2007]

# List of main algorithms of clustering

## ▶ **Partitioning methods:**

$k$ -means,  $k$ -medians,  $k$ -centers,  $k$ -medoids

## ▶ **Hierarchical methods:** linkages-based clustering (AHC)

BIRCH: Balanced Iterative Reducing and Clustering using Hierarchies [Zhang et al., 1996]

CURE: Clustering Using REpresentatives [Guha et al., 1998]

ROCK: RObust Clustering using linkS [Guha et al., 1999]

CHAMELEON: dynamic model to measure similarity of clusters [Karypis et al., 1999]

## ▶ **Arbitrarily shaped clusters:**

DbScan: Density-based clustering [Ester, 1996]

OPTICS: Ordering Points To Identify the Clustering Structure [Ankerst et al., 1999]

## ▶ **Model-based methods:**

Naive-Bayes model [Meila and Heckerman, 2001]

Mixture of Gaussian models [Banfield and Raftery, 1993]

Neural network (SOM, Self-Organizing Map) [Kohonen, 1981]

## ▶ **Spectral clustering methods** [Ng et al., 2001]

a recent method based on algebraic process of squared distance matrix

# Clustering vs Classification

NIPS 2005,2009 workshop on Theoretical Foundations of Clustering  
Shai Ben-David, Ulrike von Luxburg, John Shawe-Taylor, Naftali Tishby

	<b>Classification</b>	<b>Clustering</b>
K	classes (given)	clusters (unknown)
Quality	Generalization error	many cost functions
Focus on	Test set	Training set
Goal	Prediction	Interpretation
Analysis	discriminant	exploratory
Field	mature	new

# Open questions of clustering

- ▶ The number of clusters

*k*-means, *k*-median, *k*-center, *k*-medoids                      set by user

Model-based method    determined by user

Affinity Propagation    indirectly set by user

- ▶ Optimality w.r.t. distortion

- ▶ Generalization property: stability w.r.t. the data sample/distribution

# Open questions of clustering

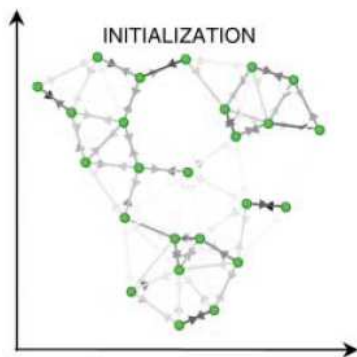
- ▶ The number of clusters
  - k*-means, *k*-median, *k*-center, *k*-medoids set by user
  - Model-based method determined by user
  - Affinity Propagation indirectly set by user
- ▶ Optimality w.r.t. distortion
- ▶ Generalization property: stability w.r.t. the data sample/distribution

## Affinity Propagation (AP)

[Frey and Dueck, 2007]

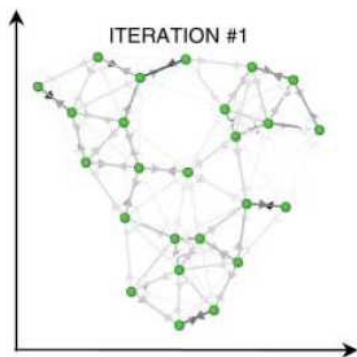
# Iterations of Message passing in AP

non-exemplar  exemplar



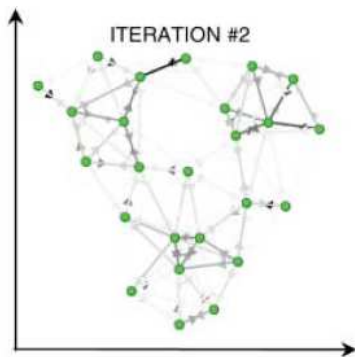
# Iterations of Message passing in AP

non-exemplar  exemplar



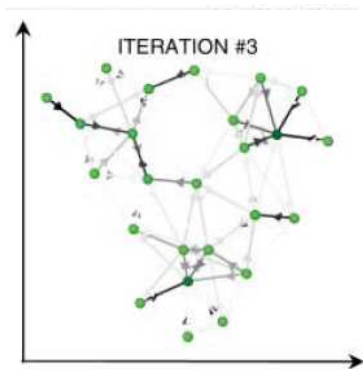
# Iterations of Message passing in AP

non-exemplar  exemplar



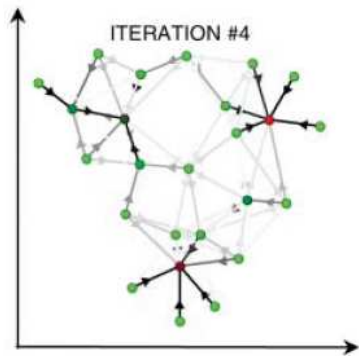
# Iterations of Message passing in AP

non-exemplar  exemplar



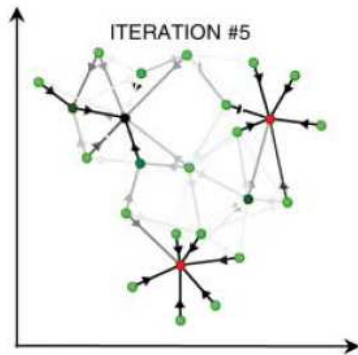
# Iterations of Message passing in AP

non-exemplar  exemplar



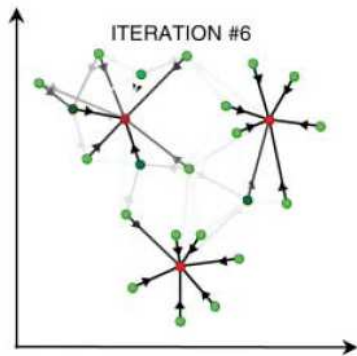
# Iterations of Message passing in AP

non-exemplar  exemplar



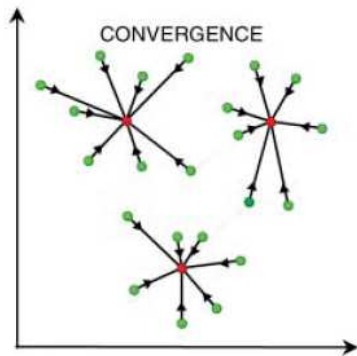
# Iterations of Message passing in AP

non-exemplar  exemplar



# Iterations of Message passing in AP

non-exemplar  exemplar



# The AP framework

input:

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

find:

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

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

where,

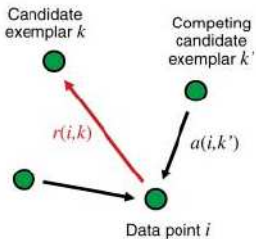
$$S(x_i, x_j) = -d^2(x_i, x_j) \quad \text{if } i \neq j$$

$$S(x_i, x_i) = -s^* \quad s^* \geq 0: \text{ user-defined parameter}$$

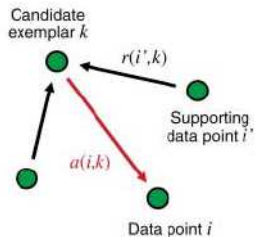
- ▶  $s^* = \infty$ , only one exemplar (one cluster)
- ▶  $s^* = 0$ , every point is an exemplar (N clusters)

# AP: a message passing algorithm

Sending responsibilities

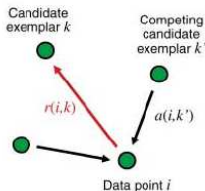


Sending availabilities

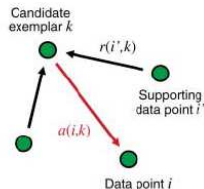


# Message passed

Sending responsibilities



Sending availabilities



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

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

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

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

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

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

## Affinity Propagation (AP)

- ▶ An exemplar-based clustering method
- ▶ A message passing algorithm (belief propagation)
- ▶ Parameterized by  $s^*$  (not by  $K$ )

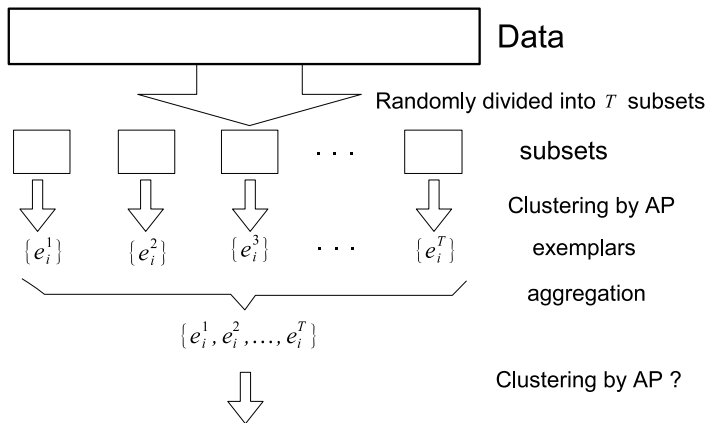
## Computational complexity

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

- ▶ Motivations
- ▶ **Clustering:**
  - The State of the Art
  - Large-scale Data Clustering
- ▶ **Streaming:** Data streams Clustering
- ▶ **Application** to Autonomic Computing:
  - A Multi-scale Real-time Grid Monitoring System
- ▶ Conclusions and Perspectives

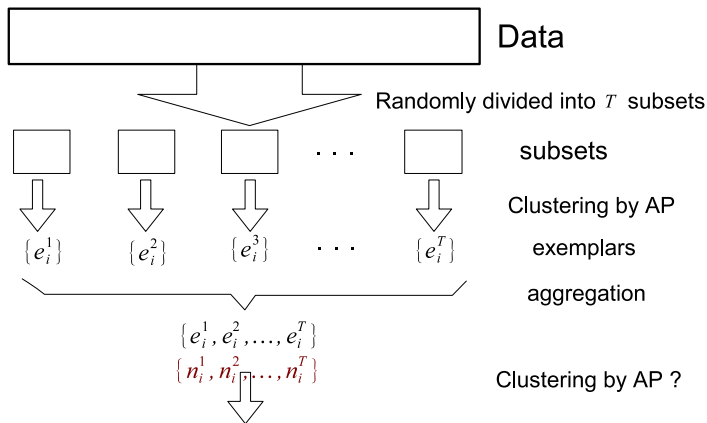
# Hierarchical AP

Divide-and-conquer (inspired by [Nittel et al., 2004])



# Hierarchical AP

Divide-and-conquer (inspired by [Nittel et al., 2004])



AP

$x_i$

$S(x_i, x_j)$

price for  $x_i$  to select  $x_j$  as an exemplar

$S(x_i, x_i)$

price to select  $x_i$  as exemplar

WAP

$x_i, n_i$

$n_i \times S(x_i, x_j)$

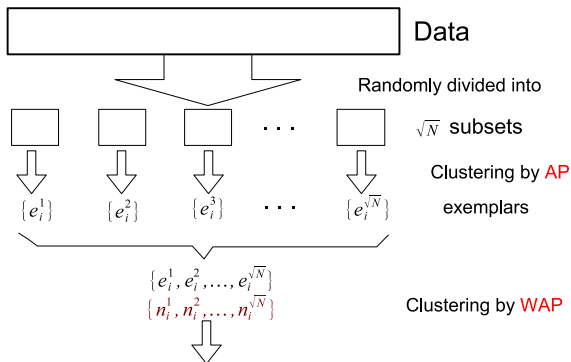
$S(x_i, x_i) + (n_i - 1) \times \epsilon$

$\epsilon$  is variance of  $n_i$  points

## Theorem

$$AP(\underbrace{x_1, \dots, x_1}_{n_1 \text{ copies}}, \underbrace{x_2, \dots, x_2}_{n_2 \text{ copies}}, \dots) == WAP((x_1, n_1), (x_2, n_2), \dots)$$

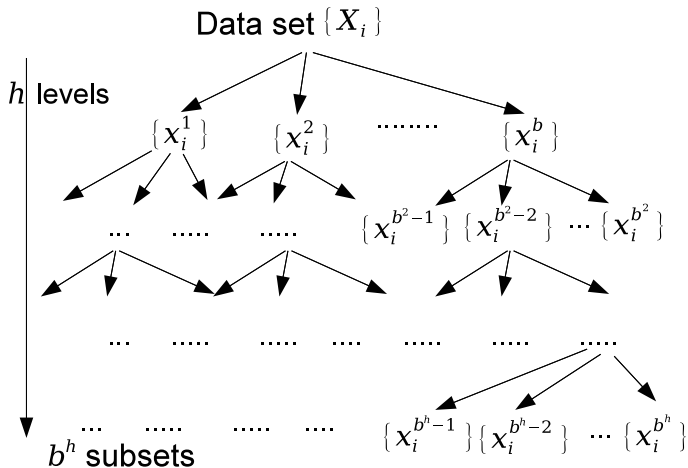
# HI-AP: Hierarchical AP



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

[Zhang et al., 2008]

# HI-AP: Hierarchical AP



- $b$ : branching factor
- $h$ : number of levels

# Complexity of HI-AP

## Theorem

HI-AP reduces the complexity to  $\mathcal{O}(N^{\frac{h+2}{h+1}})$

[Zhang et al., 2009]

$K$  : number of exemplars to be clustered on average

$b = (\frac{N}{K})^{\frac{1}{h+1}}$  : branching factor

$K^2 (\frac{N}{K})^{\frac{2}{h+1}}$  : complexity on each branching

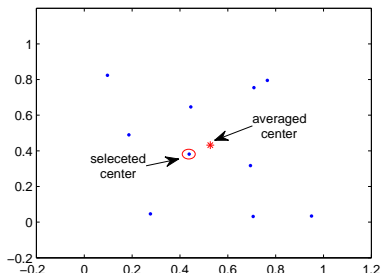
$\sum_{i=0}^h b^i = \frac{b^{h+1}-1}{b-1}$  : total number of branching

Therefore: total computational complexity:

$$C(h) = K^2 \left(\frac{N}{K}\right)^{\frac{2}{h+1}} \frac{\frac{N}{K} - 1}{\left(\frac{N}{K}\right)^{\frac{1}{h+1}} - 1} \underset{N \gg K}{\approx} K^2 \left(\frac{N}{K}\right)^{\frac{h+2}{h+1}}.$$

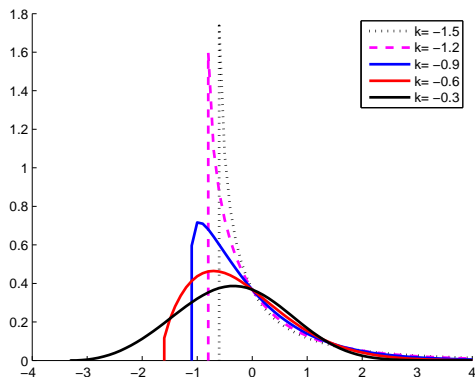
Particular cases,  $C(0) = N^2$  and  $C(1) \propto N^{3/2}$

# Study of the distortion loss



- ▶ real center of data distribution  $N(\mu, \sigma^2)$ :  $\mu$
- ▶ empirical center of  $n$  data samples:  $\hat{\mu}_n$
- ▶ distance distribution  
$$x_i - \hat{\mu}_n \sim N(0, \sigma^2 + \frac{\sigma^2}{n})$$
- ▶ selected center (exemplar) :  $\bar{\mu}_n$  (closest to  $\hat{\mu}_n$ )
- ▶ distance distribution  
$$|\bar{\mu}_n - \hat{\mu}_n| = \min(|x_i - \hat{\mu}_n|)$$
  
 $\sim$  Weibull distribution (Type III extreme value distribution)

# Weibull distribution (Type III extreme value distribution)



where  $k$  is the shape parameter.

# Cumulative radial distribution of exemplars

$$\lim_{M \rightarrow \infty} F_{sd}^{(h+1)}\left(\frac{x}{M^{(h+1)\gamma}}\right) = \begin{cases} \frac{\Gamma(\frac{d}{2}, \frac{x}{\sigma^{(h+1)}})}{\Gamma(\frac{d}{2})} & d < 2, \gamma = 1 \\ \exp(-\alpha^{(h+1)} x^{\frac{d}{2}}) & d > 2, \gamma = \frac{2}{d} \\ \exp(-\alpha^{(h+1)} x) & d = 2, \gamma = 1. \end{cases}$$

where,  $h$  is the level of HI-AP,

$F_{sd}^{(h+1)}$  is the cumulative distribution of samples at  $h$ -level,  
radial distribution of exemplars

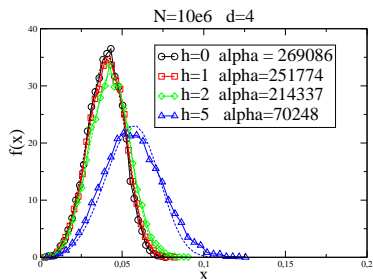
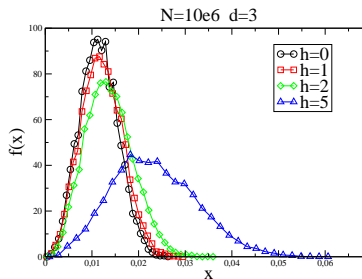
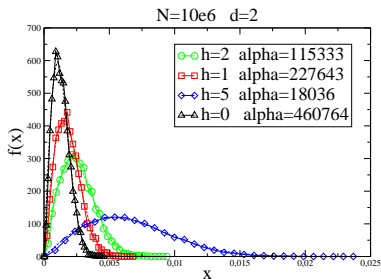
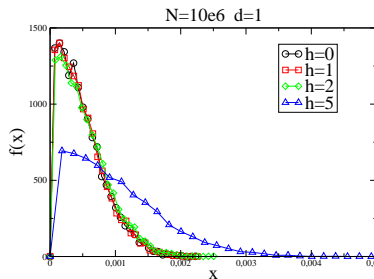
$M$  is the number of samples,

$d$  is the dimension,

$$\alpha = -\lim_{x \rightarrow 0} \frac{\log(F_{sd}(x))}{x^{\frac{d}{2}}}$$

(X. Zhang et al, SIGKDD 2009)

# Radial distribution of exemplars on different $h$ and $d$



# Validation of HI-AP on benchmark data

## Evaluation: Averaged Distortion

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

- ▶ Computational **complexity** is **reduced**
- ▶ **Limited distortion increase**

Data	K	N	D	AP	HI-AP	increased
Face (all)	14	2250	131	<b>81.45</b>	<b>84.17</b>	3.34%
Swedish Leaf	15	1125	128	<b>16.96</b>	<b>17.94</b>	5.78%

Clustering benchmark data from Eamonn Keogh  
[www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/)

- ▶ Motivations
- ▶ **Clustering:**
  - The State of the Art
  - Large-scale Data Clustering
- ▶ **Streaming:** Data streams Clustering
- ▶ **Application** to Autonomic Computing:
  - A Multi-scale Real-time Grid Monitoring System
- ▶ Conclusions and Perspectives

# Streaming: the state of the art



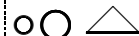
Model



Model



Model



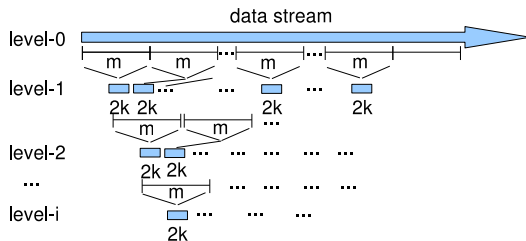
## Open questions:

- ▶ Model available any time
- ▶ How to deal with changes
- ▶ Quality of the model (distortion + size of the models)

## Sliding window strategy

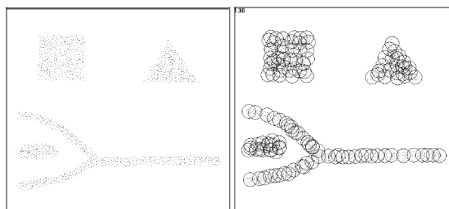
[Guha et al., 2000]

fixed segmentation window  $\text{---} >$  hinders the catching of the distribution changes



## A two-level scheme

[Aggarwal et al., 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 method DBScan [Cao et al., 2006]

**Limitation:** Model only available upon request.

# Extending AP to data streaming

## Goal:

- ▶ providing an online summary made of exemplars
- ▶ coping with non-stationary distribution

## STRAP:

- ▶ combine AP with change detection test
- ▶ self-adapt change detection test parameters

# STRAP: Extending AP to data streaming



Model



Reservoir



# STRAP: Extending AP to data streaming



Does  $x_t$  fit the current model ?

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

# STRAP: Extending AP to data streaming



Does  $x_t$  fit the current model ?

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

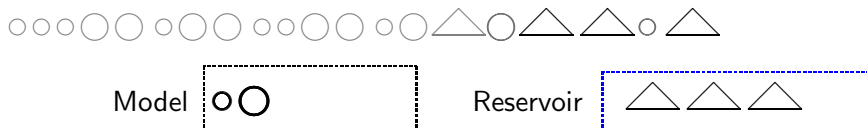
# STRAP: Extending AP to data streaming



Does  $x_t$  fit the current model ?

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

# STRAP: Extending AP to data streaming

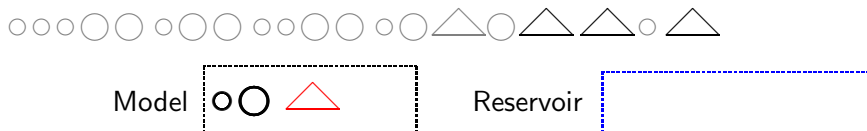


Has the distribution changed ?

CHANGE TEST

- ▶ if yes, rebuild the model
- ▶ otherwise, continue

# STRAP: Extending AP to data streaming

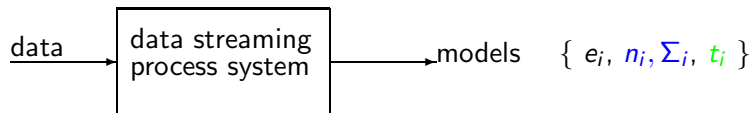


Has the distribution changed ?

CHANGE TEST

- ▶ if yes, rebuild the model
- ▶ otherwise, continue

# STRAP Method



Does  $x_t$  fit the current model ?

- ▶ if yes, **update the model** update the weight with time decay
- ▶ otherwise, go to reservoir

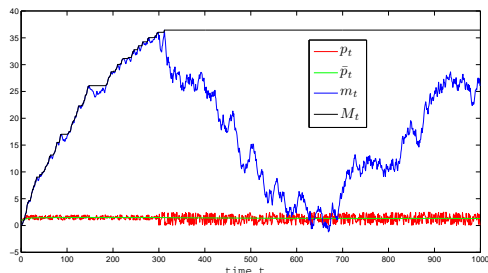
Has the distribution changed ?

- ▶ if yes, **rebuild 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, 1954, Hinkley, 1971]



$p_t$  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 the threshold  $\lambda$  ?

# Setting of threshold $\lambda$

- ▶ fixed empirical value [Zhang et al., 2008]
- ▶ self-adaptive change detection test [Zhang et al., 2009]

**Self-adapt  $\lambda$**   $\equiv$  An optimization problem

Optimization criterion: Bayesian Information Criterion

[Schwarz, 1978]

**BIC**:  $\mathcal{F}_\lambda =$

$$\sum_{i=1}^{|C|} (\sum_{e_j \in C_i} d(x_j, e_j)) \quad \text{loss}$$
$$+ \rho \log N \quad \text{size of model}$$
$$+ \eta O_t \quad \text{percentage of outliers}$$

## OPTIMIZATION:

- ▶  $\epsilon$ -greedy search from a finite set of  $\lambda$  values

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

$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	...
$\mathbf{E}(F_{\lambda_1})$	$\mathbf{E}(F_{\lambda_2})$	$\mathbf{E}(F_{\lambda_3})$	$\mathbf{E}(F_{\lambda_4})$	...

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

# 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., 2006]

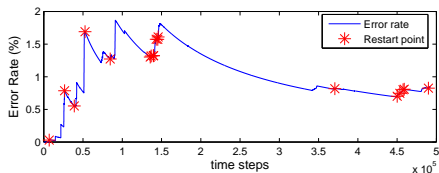
## Performance indicator (supervised setting)

- ▶ Clustering accuracy =  $\frac{\sum_{i=1}^K |C_i^e|}{N}$
- ▶ Clustering purity =  $\frac{1}{K} \sum_{i=1}^K \frac{|C_i^d|}{|C_i|}$

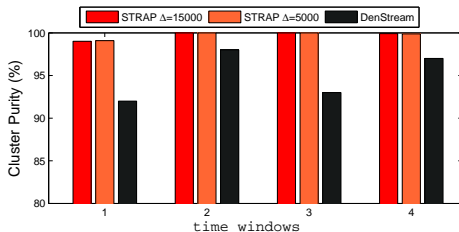
KDD Cup 1999 data: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.

# Accuracy and Purity along time

Error Rate along time  $< 2\%$



Higher clustering purity than DenStream



## 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

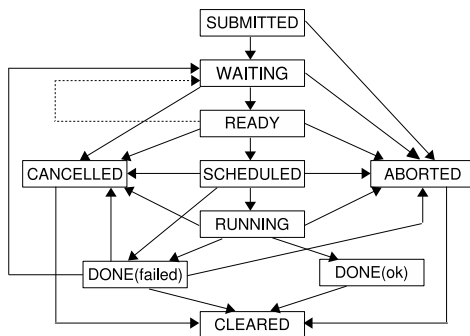
- ▶ *DenStream*: 7 seconds

- ▶ STRAP : 7 mins

slower by one order of magnitude  
due to the model available at any time

- ▶ Motivations
- ▶ **Clustering:**
  - The State of the Art
  - Large-scale Data Clustering
- ▶ **Streaming:** Data streams Clustering
- ▶ **Application to Autonomic Computing:**
  - A Multi-scale Real-time Grid Monitoring System
- ▶ Conclusions and Perspectives

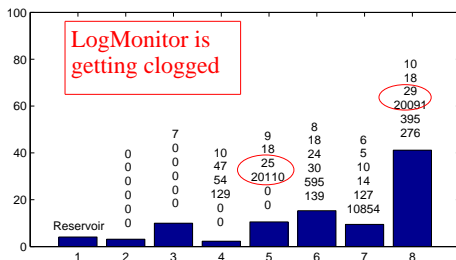
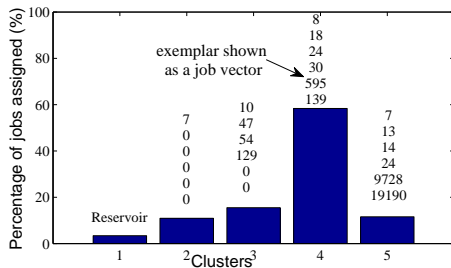
# EGEE streaming jobs



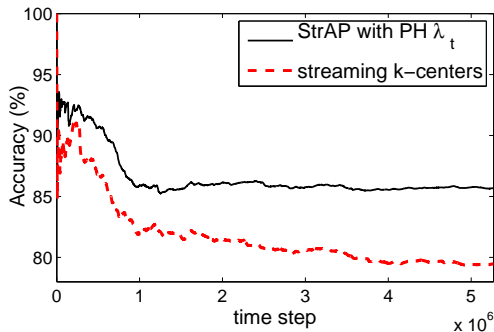
- ▶ EGEE logs of 39 RBs during 5 months (2006-01-01 ~ 2006-05-31)
- ▶ 5,268,564 jobs
- ▶ for each job, its
  - ▶ final status (successful or failure)
  - ▶ **6 features** describing the **time-cost** of services in a job lifecycle

# Real-time Monitoring: summarizing the job stream

Online summarizing the streaming jobs into clusters:



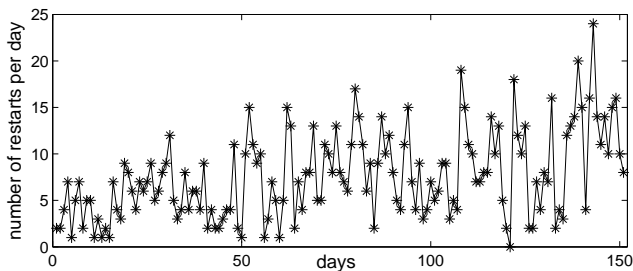
# Clustering Accuracy



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

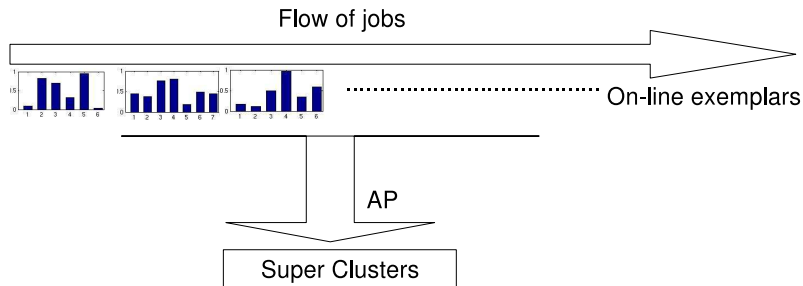
# Discussion

- ▶ Real-time quality (330K jobs/day):
  - ▶ tested on Intel 2.66GHz Dual-Core PC with 2 GB memory
  - ▶ **60k jobs per minute** C/C++
- ▶ **Concise online summary** of the streaming jobs, with
  - ▶ **proportion** of failures
  - ▶ performance of the grid services
- ▶ **Dynamics** of the load distribution

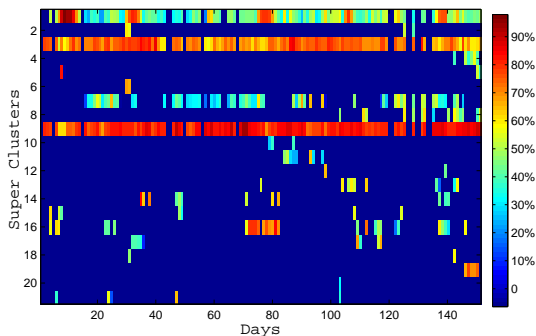


# Large-time scale offline analysis

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



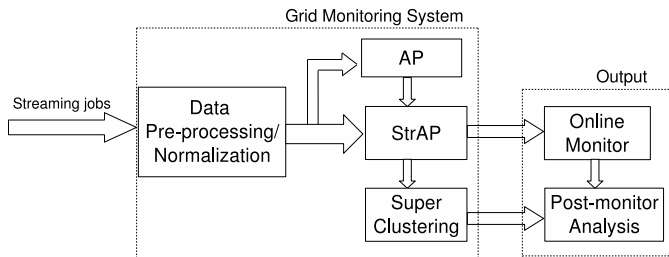
# Excerpt of the general history (failures)



“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

# G-strap: Multi-scale Real-time Grid Monitoring System



- ▶ provide **multi-scale** models to describe the Grid status
- ▶ guarantee the **quality of model** w.r.t the optimal exemplars and accuracy
- ▶ cope with the **non-stationary** distribution

- ▶ Motivations
- ▶ **Clustering:**
  - The State of the Art
  - Large-scale Data Clustering
- ▶ **Streaming:** Data streams Clustering
- ▶ **Application** to Autonomic Computing:
  - A Multi-scale Real-time Grid Monitoring System
- ▶ **Conclusions and Perspectives**

## Algorithms:

- ▶ Extending AP to a quasi-linear algorithm HI-AP
- ▶ Analyzing the distortion loss
- ▶ Extending AP to data streaming STRAP
- ▶ Self-tuned adaption to non-stationary distribution
- ▶ Guaranteeing the performance w.r.t distortion and discrimination accuracy

## Grid modeling: G-STRAP

- ▶ Model available any time, real-time
- ▶ Multi-scale support for the system administrator

Fixed number of clusters by messaging passing

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

AP with given  $K$  :

$S(x_i, x_j) \leftarrow$  by messages *Responsibility* and *Availability*  
with the constraint:  
the number of exemplars =  $K$

## Application-wise

- ▶ more complex job description
- ▶ toward user profiling (user-friendly help)
- ▶ coupling with alarm system
- ▶ exploiting the empirical distribution to support optimal scheduling

- ▶ **SIGKDD'2009**    **X. Zhang**, C. Furtlehner, J. Perez, C. Germain, M. Sebag, "Toward Autonomic Grids: Analyzing the Job Flow with Affinity Streaming".
- ▶ **ECML/PKDD'2008**    **X. Zhang**, C. Furtlehner, M. Sebag, "Data streaming with Affinity propagation".
- ▶ **CCGrid'2009**    **X. Zhang**, M. Sebag, C. Germain, "Multi-scale Realtime Grid Monitoring with Job Stream Mining".
- ▶ **CAp'2009**    **X. Zhang**, C. Furtlehner, C. Germain, M. Sebag, "G-StrAP: A 2-level Real-time Grid Monitoring System".
- ▶ **EGEE User Forum'2009**    **X. Zhang**, C. Furtlehner, C. Germain, M. Sebag, "Grid Monitoring by Online Clustering".
- ▶ **STAIRS'2008**    **X. Zhang**, C. Furtlehner, M. Sebag, "Distributed and Incremental Clustering Based on Weighted Affinity Propagation".
- ▶ **CAp'2008**    **X. Zhang**, C. Furtlehner, M. Sebag, "Frugal and online affinity propagation".
- ▶ **RFIA'2008**    **X. Zhang**, M. Sebag, C. Germain, "Modelling the jobs of a Grid System".
- ▶ **ICDMW'2007**    **X. Zhang**, M. Sebag, C. Germain, "Toward Behavioral Modeling of a Grid System: Mining the Logging and Bookkeeping files".

## References:



Aggarwal, C. C., Han, J., Wang, J., and Yu, P. S. (2003).

A framework for clustering evolving data streams.

In *VLDB*, pages 81–92.



Ankerst, M., Breunig, M. M., Peter Kriegel, H., and Sander, J. (1999).

OPTICS: Ordering points to identify the clustering structure.

In *SIGMOD Conference*, pages 49–60.



Banfield, J. D. and Raftery, A. E. (1993).

Model-based gaussian and non-gaussian clustering.

*Biometrics*, 49:803–821.



Bradley, P. S., Mangasarian, O. L., and Street, W. N. (1997).

Clustering via concave minimization.

In *Advances in Neural Information Processing Systems (NIPS)*, pages 368–374.



Cao, F., Ester, M., Qian, W., and Zhou, A. (2006).

Density-based clustering over an evolving data stream with noise.

In *SIAM Conference on Data Mining (SDM)*, pages 326–337.



Ester, M. (1996).

A density-based algorithm for discovering clusters in large spatial databases with noise.

In *Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–231.



Frey, B. J. and Dueck, D. (2007).

Clustering by passing messages between data points.

*Science*, 315:972–976.



Guha, S., Mishra, N., Motwani, R., and O’Callaghan, L. (2000).

Clustering data streams.

In *IEEE Symposium on Foundations of Computer Science*, pages 359–366.



Guha, S., Rastogi, R., and Shim, K. (1998).

CURE: an efficient clustering algorithm for large databases.

In *SIGMOD'98: Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 73–84.



Guha, S., Rastogi, R., and Shim, K. (1999).

ROCK: A robust clustering algorithm for categorical attributes.

In *ICDE '99: Proceedings of the 15th International Conference on Data Engineering*, pages 512–521.



Hinkley, D. (1971).

Inference about the change-point from cumulative sum tests.

*Biometrika*, 58:509–523.



Karypis, G., Han, E.-H., and Kumar, V. (1999).

CHAMELEON: A hierarchical clustering algorithm using dynamic modeling.

*Computer*, 32:68–75.



Kaufman, L. and Rousseeuw, P. (1987).

Clustering by means of medoids.

In *Statistical Data Analysis Based on the L1 Norm and Related Methods*, pages 405–416.



Kaufman, L. and Rousseeuw, P. (1990).

*Finding Groups in Data: an introduction to cluster analysis*.

Wiley.



Kohonen, T. (1981).

Automatic formation of topological maps of patterns in a self-organizing system.

In *Proceedings of the 2nd Scandinavian Conference on Image Analysis*, pages 214–220.



Meila, M. and Heckerman, D. (2001).

An experimental comparison of model-based clustering methods.

*Machine Learning*, 42(1/2):9–29.



Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001).

On spectral clustering: Analysis and an algorithm.

In *Advances in Neural Information Processing Systems (NIPS)*, pages 849–856.



Ng, R. and Han, J. (1994).

Efficient and effective clustering methods for spatial data mining.  
In *VLDB*, pages 144–155.



Nittel, S., Leung, K. T., and Braverman, A. (2004).

Scaling clustering algorithms for massive data sets using data streams.  
In *ICDE '04*.



Page, E. (1954).

Continuous inspection schemes.  
*Biometrika*, 41:100–115.



Rish, I., Brodie, M., and et al, S. M. (2005).

Adaptive diagnosis in distributed systems.  
*IEEE Transactions on Neural Networks*, 16:1088–1109.



Schwarz, G. (1978).

Estimating the dimension of a model.  
*The Annals of Statistics*, 6:461–464.



Zhang, T., Ramakrishnan, R., and Livny, M. (1996).

BIRCH: an efficient data clustering method for very large databases.  
In *SIGMOD'96: Proceedings of ACM SIGMOD international conference on Management of data*, pages 103–114.



Zhang, X., Furtlehner, C., Perez, J., Germain-Renaud, C., and Sebag, M. (2009).

Toward autonomic grids: Analyzing the job flow with affinity streaming.  
In *ACM SIGKDD*, pages 987–995.



Zhang, X., Furtlehner, C., and Sebag, M. (2008).

Data streaming with affinity propagation.  
In *ECML/PKDD*, pages 628–643.