Master Recherche IAC
Option 2
Apprentissage Statistique & Optimisation Avancés

Anne Auger – Balazs Kégl – Michèle Sebag
LRI – LAL

Feb. 5th, 2014

Clustering

Input

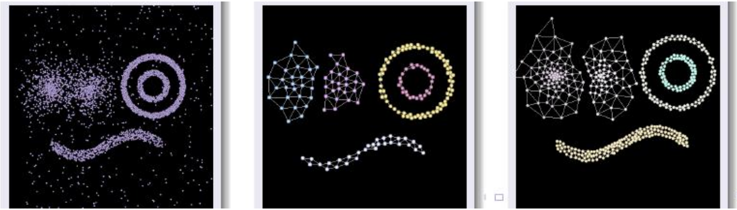
$$\mathcal{E} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \sim P(x)$$

Output

- Models $\hat{P}(x)$
- Clusters Partition
- Representatives

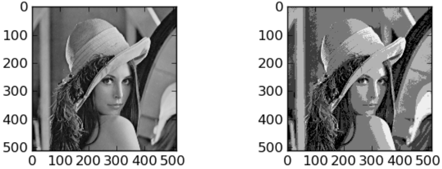
Assumptions, contexts

Clusters are separated by a low-density region



Motivations

- Compression.
  Ex, vector quantization in images.



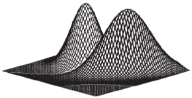
- Divide and conquer; preliminary for classification.
  Ex, different types of diseases.
- Check data.

Overview

Clustering Questions

Hard or soft ?

- **Hard**: find a partition of the data
- **Soft**: estimate the distribution of the data as a mixture of components.



Parametric vs non Parametric ?

- **Parametric**: number K of clusters is known
- **Non-Parametric**: find K
  (wrapping a parametric clustering algorithm)

Caveat:

- Complexity
- Outliers
- Validation

Formal Background

Notations

$\mathcal{E}$	$\{\mathbf{x}_1, \ldots \mathbf{x}_N\}$ dataset	
N	number of data points	
K	number of clusters	given or optimized
C_k	k-th cluster	**Hard clustering**
$\tau(i)$	index of cluster containing $\mathbf{x}_i$	
f_k	k-th model	**Soft clustering**
$\gamma_k(i)$	$Pr(\mathbf{x}_i \sim f_k)$	

Solution	Hard Clustering	Partition $\Delta = (C_1, \ldots C_k)$
	Soft Clustering	$\forall i \ \sum_k \gamma_k(i) = 1$

Formal Background, 2

Quality / Cost function Measures how well the clusters characterize the data

- (log)likelihood soft clustering
- dispersion hard clustering

$$\sum_{k=1}^{K} \frac{1}{|C_k|^2} \sum_{\mathbf{x}_i, \mathbf{x}_j \ in \ C_k} d(\mathbf{x}_i, \mathbf{x}_j)^2$$

Formal Background, 2

Quality / Cost function Measures how well the clusters characterize the data

- (log)likelihood — soft clustering
- dispersion — hard clustering

$$\sum_{k=1}^{K} \frac{1}{|C_k|^2} \sum_{\mathbf{x}_i, \mathbf{x}_j \ in \ C_k} d(\mathbf{x}_i, \mathbf{x}_j)^2$$

Tradeoff Quality increases with $K \Rightarrow$ Regularization needed
to avoid one cluster per data point

Exercize

$$\sum_{k=1}^{K} \frac{1}{|C_k|^2} \sum_{\mathbf{x}_i, \mathbf{x}_j \ in \ C_k} ||\mathbf{x}_i - \mathbf{x}_j||^2 = \sum_{k=1}^{K} \frac{1}{|C_k|^2} \sum_{\mathbf{x}_i, \mathbf{x}_j \ in \ C_k} ||\mathbf{x}_i - \bar{\mathbf{x}}_k||^2$$

with $\bar{\mathbf{x}}_k =$ average $\mathbf{x}_i, \mathbf{x}_i \in C_k$.

Clustering vs Classification

Marina Meila http://videolectures.net/

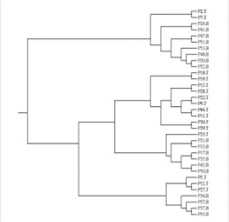
	Classification	Clustering
K	# classes (given)	# clusters (unknown)
Quality	Generalization error	many cost functions
Focus on	Test set	Training set
Goal	Prediction	Interpretation
Analysis	discriminant	exploratory
Field	mature	new

Non-Parametric Clustering

Hierarchical Clustering

Principle

- agglomerative (join nearest clusters)
- divisive (split most dispersed cluster)



Algorithm

Init: Each point is a cluster (n clusters)

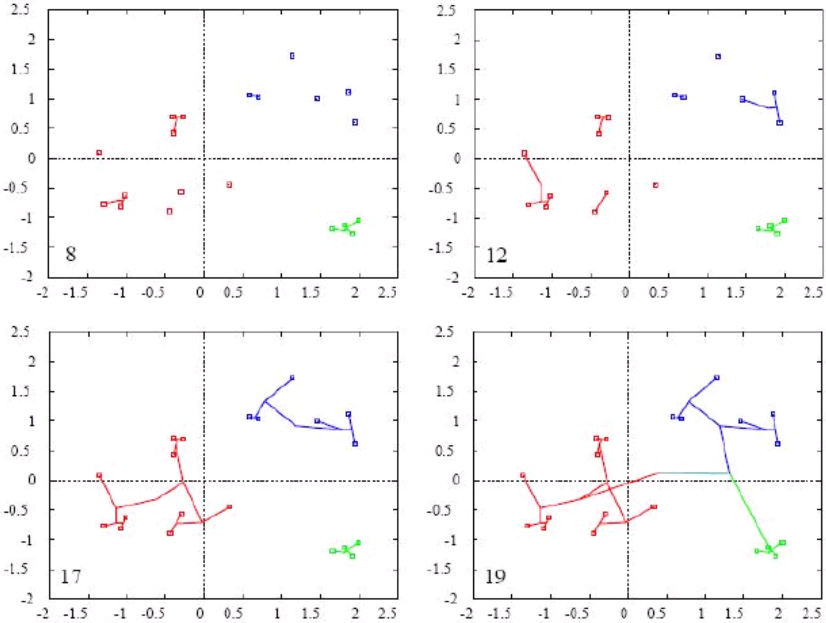
Loop

  Select two most similar clusters

  Merge them

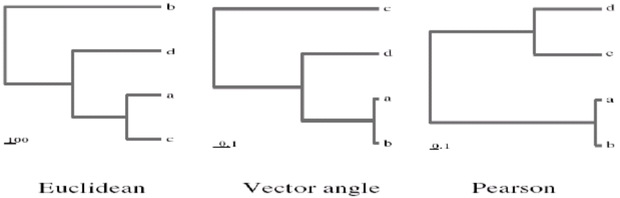
Until there is only 1 cluster

Hierarchical Clustering, example



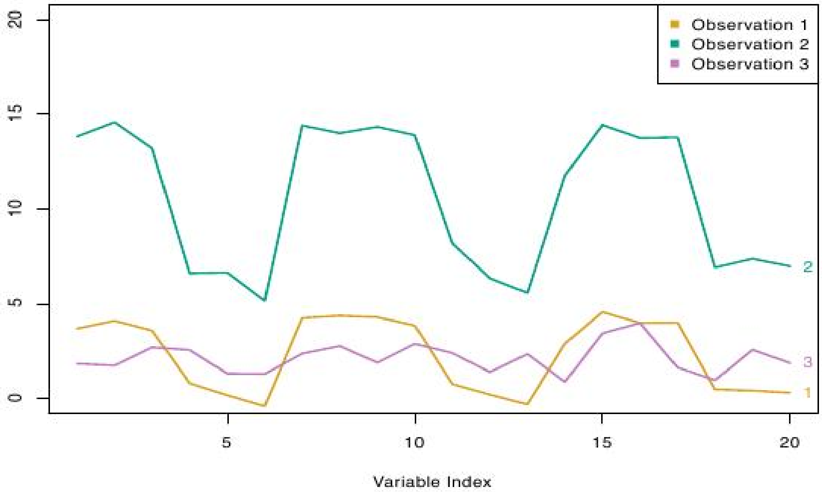
Hierarchical Clustering, 2

Key point 1: choice of distance



$$d(x, x') = \begin{cases} \sqrt{\sum_i (x_i - x_i')^2} & \text{Euclidean distance} \\ 1 - \frac{\sum_i x_i x_i'}{||x||.||x'||} & \text{Cosine angle} \\ 1 - \frac{\sum_i (x_i - \bar{x})(x_i' - \bar{x}')}{||x - \bar{x}||.||x' - \bar{x}'||} & \text{Pearson} \end{cases}$$

Hierarchical Clustering, 3



Hierarchical Clustering, 4

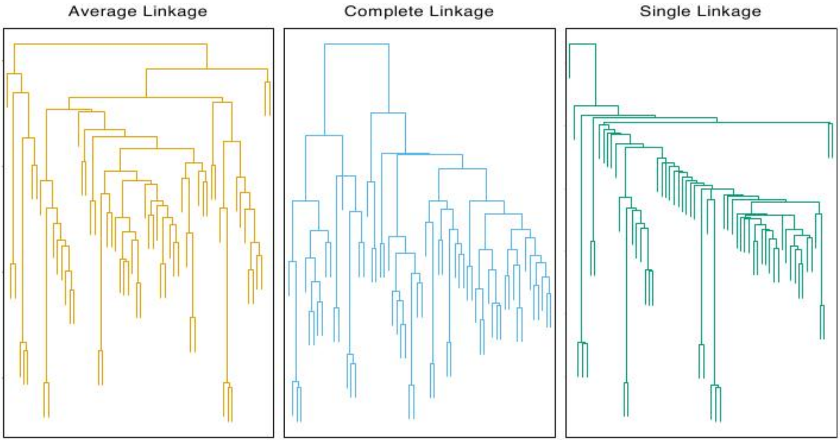
Key point 2: choice of aggregation

Compute distance between two clusters

- Complete linkage:Largest distance between points
- Single linkage: Smallest distance between points
- Average linkage: Average distance between points
- Centroid: distance between centroids of the points

Centroid of points: point closest to their average.

Hierarchical Clustering, 5



Parametric Clustering

Parametric: K is known

Algorithms based on distances

- K-means
- graph / cut

Algorithms based on models

- Mixture of models: EM algorithm

Overview

Clustering
- K-Means
- Generative models
- Expectation Maximization
- Selecting the number of clusters
- Stability

Axiomatisation

Data Streaming

Application: Autonomic Computing

Learning distances

K-Means

Algorithm

1. Init:
   Uniformly draw K points $\mathbf{x}_{i_j}$ in $\mathcal{E}$
   Set $C_j = \{\mathbf{x}_{i_j}\}$
2. Repeat
3. Draw without replacement $\mathbf{x}_i$ from $\mathcal{E}$
4. $\tau(i) = argmin_{k=1\ldots K}\{\mathbf{d}(\mathbf{x_i}, \mathbf{C_k})\}$ **find best cluster for $\mathbf{x}_i$**
5. $C_{\tau(i)} = C_{\tau(i)} \bigcup \mathbf{x}_i$ **add $\mathbf{x}_i$ to $C_{\tau(i)}$**
6. Until all points have been drawn
7. If partition $C_1 \ldots C_K$ has changed **Stabilize**
   Define $\mathbf{x}_{i_k} =$ **best point** in C_k, $C_k = \{x_{i_k}\}$, goto 2.

Algorithm terminates

K-Means, Knobs

Knob 1 : define $d(\mathbf{x}_i, C_k)$	**favors**
▶ $min\{d(\mathbf{x}_i, \mathbf{x}_j), \mathbf{x}_j \in C_k\}$	long clusters
▶ $average\{d(\mathbf{x}_i, \mathbf{x}_j), \mathbf{x}_j \in C_k\}$	compact clusters
▶ $max\{d(\mathbf{x}_i, \mathbf{x}_j), \mathbf{x}_j \in C_k\}$	spheric clusters

Knob 2 : define "best" in C_k

- ▶ Medoid: $argmin_i\{\sum_{\mathbf{x}_j \in C_k} d(\mathbf{x}_i, \mathbf{x}_j)\}$
- * Average (does not belong to $\mathcal{E}$): $\frac{1}{|C_k|} \sum_{\mathbf{x}_j \in C_k} \mathbf{x}_j$

No single best choice

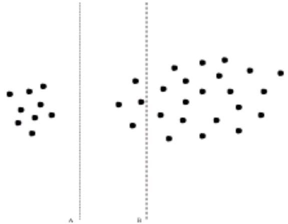


FIG. 1. Optimizing the diameter produces B while A is clearly more desirable.

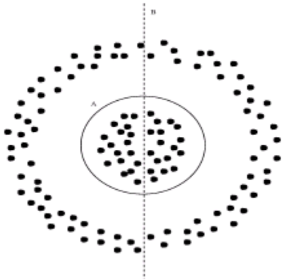


FIG. 2. The inferior clustering B is found by optimizing the 2-median measure.

K-Means, Discussion

PROS

- **Complexity** $\mathcal{O}(K \times N)$
- Can incorporate prior knowledge initialization

CONS

- Sensitive to initialization
- Sensitive to outliers
- Sensitive to irrelevant attributes

K-Means, Convergence

- For cost function

$$\mathcal{L}(\Delta) = \sum_k \sum_{i,j \ / \ \tau(i)=\tau(j)=k} d(\mathbf{x}_i, \mathbf{x}_j)$$

- for $d(\mathbf{x}_i, C_k) =$ average $\{d(\mathbf{x}_i, \mathbf{x}_j), \mathbf{x}_j \in C_k\}$
- for "best" in C_k = average of $\mathbf{x}_j \in C_k$

K-means converges toward a (local) minimum of $\mathcal{L}$.

K-Means, Practicalities

Initialization

- Uniform sampling
- Average of $\mathcal{E}$ + random perturbations
- Average of $\mathcal{E}$ + orthogonal perturbations
- Extreme points: select $\mathbf{x}_{i_1}$ uniformly in $\mathcal{E}$, then

$$\text{Select } x_{i_j} = argmax\{\sum_{k=1}^{j} d(\mathbf{x}_i, x_{i_k})\}$$

Pre-processing

- Mean-centering the dataset

Overview

- Clustering
  - K-Means
  - Generative models
  - Expectation Maximization
  - Selecting the number of clusters
  - Stability
- Axiomatisation
- Data Streaming
- Application: Autonomic Computing
- Learning distances

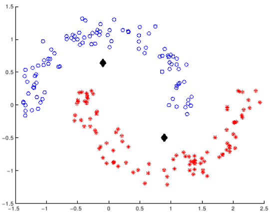
Model-based clustering

Mixture of components

- Density $f = \sum_{k=1}^{K} \pi_k f_k$
- f_k: the k-th component of the mixture
- $\gamma_k(i) = \frac{\pi_k f_k(x)}{f(x)}$
- induces $C_k = \{\mathbf{x}_j \ / \ k = argmax\{\gamma_k(j)\}\}$

Nature of components: prior knowledge

- Most often Gaussian: $f_k = (\mu_k, \Sigma_k)$
- Beware: clusters are not always Gaussian...



Model-based clustering, 2

Search space

- Solution : $(\pi_k, \mu_k, \Sigma_k)_{k=1}^K = \theta$

Criterion: log-likelihood of dataset

$$\ell(\theta) = \log(Pr(\mathcal{E})) = \sum_{i=1}^N \log Pr(\mathbf{x}_i) \propto \sum_{i=1}^N \sum_{k=1}^K \log(\pi_k f_k(\mathbf{x}_i))$$

to be maximized.

Model-based clustering with EM

Formalization

- Define $z_{i,k} = 1$ iff $\mathbf{x}_i$ belongs to C_k.
- $E[z_{i,k}] = \gamma_k(i)$ prob. $\mathbf{x}_i$ generated by $\pi_k f_k$
- Expectation of log likelihood

$$
\begin{aligned}
E[\ell(\theta)] &\propto \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_i(k) \log(\pi_k f_k(\mathbf{x}_i)) \\
&= \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_i(k) \log \pi_k + \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_i(k) \log f_k(\mathbf{x}_i)
\end{aligned}
$$

EM optimization

E step Given θ, compute

$$\gamma_k(i) = \frac{\pi_k f_k(\mathbf{x}_i)}{f(x)}$$

M step Given $\gamma_k(i)$, compute

$$\theta^* = (\pi_k, \mu_k, \Sigma_k)^* = argmin E[\ell(\theta)]$$

Maximization step

π_k: Fraction of points in C_k

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \gamma_k(i)$$

μ_k: Mean of C_k

$$\mu_k = \frac{\sum_{i=1}^{N} \gamma_k(i) \mathbf{x}_i}{\sum_{i=1}^{N} \gamma_k(i)}$$

Σ_k: Covariance

$$\Sigma_k = \frac{\sum_{i=1}^{N} \gamma_k(i)(\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)'}{\sum_{i=1}^{N} \gamma_k(i)}$$

Overview

Clustering
- K-Means
- Generative models
- Expectation Maximization
- Selecting the number of clusters
- Stability

Axiomatisation

Data Streaming

Application: Autonomic Computing

Learning distances

Choosing the number of clusters

K-means constructs a partition whatever the K value is.

Selection of K

- **Bayesian approaches**
  Tradeoff between accuracy / richness of the model
- **Stability**
  Varying the data should not change the result
- **Gap statistics**
  Compare with null hypothesis: all data in same cluster.

Bayesian approaches

Bayesian Information Criterion

$$BIC(\theta) = \ell(\theta) - \frac{\#\theta}{2} \log N$$

Select $K = \text{argmax } BIC(\theta)$
where $\#\theta$ = number of free parameters in θ:

- if all components have same scalar variance σ

$$\#\theta = K - 1 + 1 + Kd$$

- if each component has a scalar variance σ_k

$$\#\theta = K - 1 + K(d + 1)$$

- if each component has a full covariance matrix Σ_k

$$\#\theta = K - 1 + K(d + d(d - 1)/2)$$

Gap statistics

Principle: hypothesis testing

1. Consider hypothesis H_0: there is no cluster in the data. $\mathcal{E}$ is generated from a no-cluster distribution π.
2. Estimate the distribution $f_{0,K}$ of $\mathcal{L}(C_1, \ldots C_K)$ for data generated after π. Analytically if π is simple
   Use Monte-Carlo methods otherwise
3. Reject H_0 with confidence α if the probability of generating the true value $\mathcal{L}(C_1, \ldots C_K)$ under $f_{0,K}$ is less than α.

Beware: the test is done for all K values...

Gap statistics, 2

Algorithm Assume $\mathcal{E}$ extracted from a no-cluster distribution, e.g. a single Gaussian.

1. Sample $\mathcal{E}$ according to this distribution
2. Apply K-means on this sample
3. Measure the associated loss function

Repeat : compute the average $\bar{\mathcal{L}}_0(K)$ and variance $\sigma_0(K)$
Define the gap:

$$Gap(K) = \bar{\mathcal{L}}_0(K) - \mathcal{L}(C_1, \ldots C_K)$$

Rule Select min K s.t.

$$Gap(K) \geq Gap(K+1) - \sigma_0(K+1)$$

What is nice: also tells if there are no clusters in the data...

Overview

Clustering
- K-Means
- Generative models
- Expectation Maximization
- Selecting the number of clusters
- Stability

Axiomatisation

Data Streaming

Application: Autonomic Computing

Learning distances

Stability

Principle

- Consider $\mathcal{E}'$ perturbed from $\mathcal{E}$
- Construct $C'_1, \ldots C'_K$ from $\mathcal{E}'$
- Evaluate the "distance" between $(C_1, \ldots C_K)$ and $(C'_1, \ldots C'_K)$
- If small distance (stability), K is OK

Distortion $D(\Delta)$

$$\begin{array}{lll} \text{Define} \quad S & S_{ij} = & < \mathbf{x}_i, \mathbf{x}_j > \\ & (\lambda_i, v_i) & \text{i-th (eigenvalue, eigenvector) of } S \\ X & X_{i,j} = & 1 \text{ iff } \mathbf{x}_i \in C_j \end{array}$$

$$D(\Delta) = \sum_i ||\mathbf{x}_i - \mu_{\tau(i)}||^2 = tr(S) - tr(X'SX)$$

Minimal distortion $D^* = tr(S) - \sum_{k=1}^{K-1} \lambda_k$

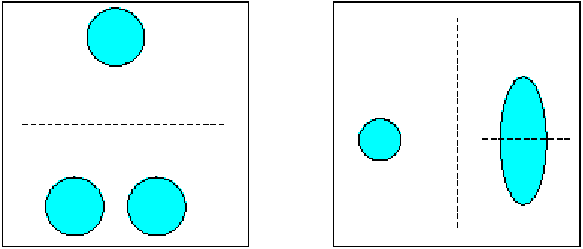
Stability, 2

Results

- Δ has low distortion $\Rightarrow (\mu_1, \ldots \mu_K)$ close to space $(v_1, \ldots v_K)$.
- Δ_1, and Δ_2 have low distortion $\Rightarrow$ "close"
- (and close to "optimal" clustering)

Meila ICML 06

Counter-example



Overview

Kleinberg's axiomatic framework for clustering

Kleinberg 2002

Given $\mathcal{E} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}_i \in X\}$, a clustering builds a partition Γ depending on distance d. Let denote $\Gamma = f(d)$.

$$\begin{pmatrix} 1 & 10 & 10 \\ 10 & 0 & 1 \\ 10 & 1 & 0 \end{pmatrix}$$

$\Gamma = (\{1\}, \{2, 3\})$.

Kleinberg's axiomatic framework for clustering

Properties

Scale invariance

$$\forall \alpha > 0, f(\alpha d) = f(d)$$

Richness

$$Range(f) = \text{ Power set of } \mathcal{E}$$

Consistency

If $f(d) = \Gamma$ and d' is a Γ-enhancing transformation of d, then

$$f(d') = \Gamma$$

where d' is Γ-enhancing if

- $d'(\mathbf{x}_i, \mathbf{x}_j) \leq d(\mathbf{x}_i, \mathbf{x}_j)$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ in same cluster of Γ
- $d'(\mathbf{x}_i, \mathbf{x}_j) \geq d(\mathbf{x}_i, \mathbf{x}_j)$ otherwise

Examples

Run single linkage till you get k clusters

- Scale invariance Yes, consistency Yes, richness No

Run single linkage while distances $\leq c \cdot max_{i,j} d(\mathbf{x}_i, \mathbf{x}_j)$, $c > 0$

Examples

Run single linkage till you get k clusters

- Scale invariance Yes, consistency Yes, richness No

Run single linkage while distances $\leq c \cdot max_{i,j} d(\mathbf{x}_i, \mathbf{x}_j)$, $c > 0$

- Scale invariance Yes, consistency No, richness Yes

Run single linkage until distances $\leq$ some threshold r

Examples

Run single linkage till you get k clusters

- Scale invariance Yes, consistency Yes, richness No

Run single linkage while distances $\leq c \cdot max_{i,j} d(\mathbf{x}_i, \mathbf{x}_j)$**,** $c > 0$

- Scale invariance Yes, consistency No, richness Yes

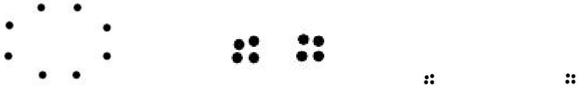
Run single linkage until distances $\leq$ **some threshold** r

- Scale invariance No, consistency Yes, richness Yes

Impossibility result

Thm

- There is no consistent way of choosing a level of granularity
- There exists no f satisfying all three axioms



d | d' enhancing Γ | d“ rescaling d'

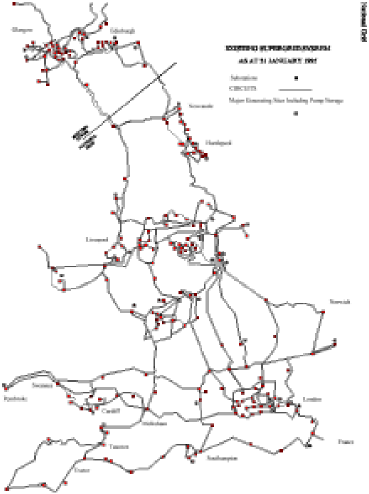
Overview

Part 2. Data Streaming

- When: data, specificities
- What: goals
- How: algorithms

More: see Joao Gama's tutorial,
http://wiki.kdubiq.org/summerschool2008/index.php/Main/Materials

Motivations



Electric Power Network

Data

Input

- Continuous flow of (possibly corrupted) data, high speed
- Huge number of sensors, variable along time (failures)
- Spatio-temporal data

Output

- Cluster: profiles of consumers
- Prediction: peaks of demand
- Monitor Evolution: Change detection, anomaly detection

Where is the problem ?

Standard Data Analysis

- Select a sample
- Generate a model (clustering, neural nets, ...)

Where is the problem ?

Standard Data Analysis

- Select a sample
- Generate a model (clustering, neural nets, ...)

Does not work...

- World is not static
- Options, Users, Climate, ... change

Specificities of data

Domain

- Radar: meteorological observations
- Satellite: images, radiation
- Astronomical surveys: radio
- Internet: traffic logs, user queries, ...
- Sensor networks
- Telecommunications

Features

- Most data never seen by humans
- Need for REAL-TIME monitoring, (intrusion, outliers, anomalies,,,)

NB: Beyond ML scope: data are not iid (independent identically distributed)

Data streaming Challenges

Maintain Decision Models in real-time

- incorporate new information
- forget old/outdated information
- detect changes and adapt models accordingly

comply with speed

Unbounded training sets

Prefer fast approximate answers...

- Approximation: Find answer with factor $1 \pm \epsilon$
- Probably correct: Pr(answer correct) $=$ 1 -δ
- PAC: ϵ, δ (Probably Approximately Correct)
- Space $\approx \mathcal{O}(1/\epsilon^2 log(1/\delta))$

Data Mining vs Data Streaming

	Traditional	**Stream**
Nr. of Passes	Multiple	Single
Processing Time	Unlimited	Restricted
Memory Usage	Unlimited	Restricted
Type of Result	Accurate	Approximate
Distributed	No	Yes

What: queries on a data stream

- Sample
- Count number of distinct values / attribute
- Estimate sliding average (number of 1's in a sliding window)
- Get top-k elements

Application: Compute entropy of the stream

$$H(x) = \sum p_i log_2(p_i)$$

useful to detect anomalies

Sampling

Uniform sampling: each one out of n examples is sampled with probability 1/n.
What if we don't know the size ?

Standard

- Sample instances at periodic time intervals
- Loss of information

Reservoir Sampling

- Create buffer size k
- Insert first k elements
- Insert i-th element with probability k/i
- Delete a buffer element at random

Limitations

- Unlikely to detect changes/anomalies
- Hard to parallelize

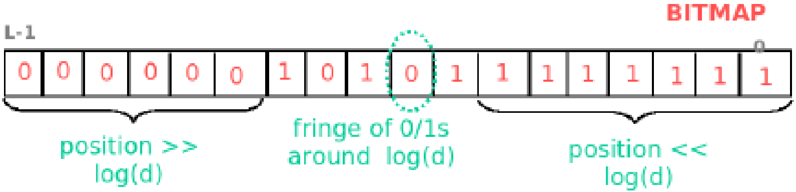
Count number of values

Problem
Domain of the attribute is $\{1, \ldots M\}$
Piece of cake if memory available... What if the memory available is $log(M)$?
Flajolet-Martin 1983
Based on hashing: $\{1, \ldots M\} \mapsto \{0, \ldots 2^L\}$ with $L = log(M)$.

$$x \rightarrow \ hash(x) = y \ \rightarrow \ position\ least\ significant\ bit, lsb(x)$$

Count number of values, followed

Init: $BITMAP(\{0,\dots L\}) = 0$
Loop: Read x, $BITMAP(lsb(x)) = 1$



Result

$$R = \text{ position of rightmost 0 in } H$$

$$M \approx 2^R/.7735$$

Decision Trees for Data Streaming

Principle

Grow the tree if evidence best attribute $>$ second best

Algorithm parameter: confidence δ (user-defined)

While true

  Read example, propagate until a leaf

  If enough examples in leaf

    Compute IG for all attributes;

    $\epsilon = \sqrt{\frac{R^2 ln(1/\delta)}{2n}}$

    Keep best if IG(best) - IG(second best) $> \epsilon$

Mining High Speed Data Streams, Pedro Domingos, Geoffrey Hulten, KDD-00

Open issues

What's new

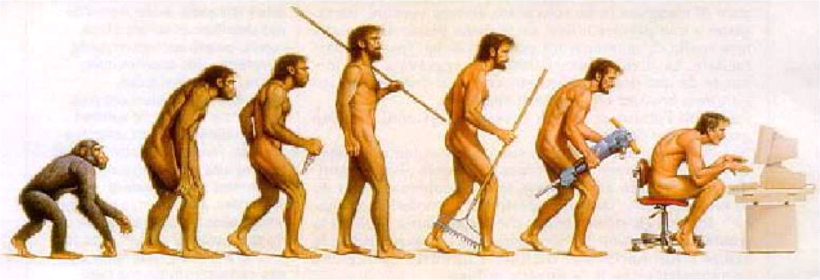
Forget about iid;
Forget about more than linear complexity (and log space)

Challenges

Online, Anytime algs
Distributed alg.
Criteria of performance
Integration of change detection

Overview

Autonomic Computing



Considering current technologies, we expect that the total number of device administrators will exceed 220 millions by 2010.

Gartner 6/2001

in Autonomic Computing Wshop, ECML / PKDD 2006

Irina Rish & Gerry Tesauro.

Autonomic Computing

The need

- Main bottleneck of the deployment of complex systems: shortage of skilled administrators

Vision

- Computing systems take care of the mundane elements of management by themselves.
- Inspiration: central nervous system (regulating temperature, breathing, and heart rate without conscious thought)

Goal

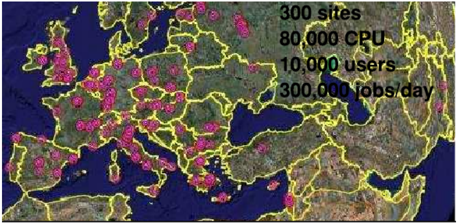
Computing systems that manage themselves in accordance with high-level objectives from humans

Kephart & Chess, IEEE Computer 2003

Toward Autonomic Grid

EGEE, Enabling Grids for E-sciencE 2001-2011

- 50 countries
- 300 sites
- 180,000 CPUs
- 5Petabytes storage
- 10,000 users
- 300,000 jobs/ day



http://public.eu-egee.org/

EGEE-III : WP Grid Observatory

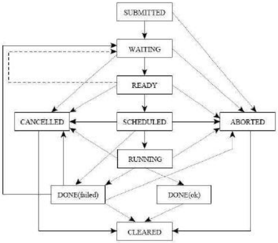
- Job scheduling
- Job profiling

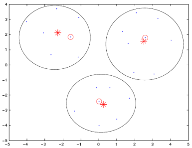
Data Streaming for Job Profiling

X. Zhang, C. Furtlehner, M.S., ECML 08; KDD 09

Position of the problem

- Jobs arrive and are processed
- Want to detect outliers and anomalies
- Want to predict the traffic / dimension the system
- The job distribution is non-stationary





Preliminary step: Clustering the jobs

Clustering with Message Passing Algorithm: Affinity Propagation

Frey and Dueck, Science 2007 **Affinity Propagation w.r.t. State of art**

	K-means	**K-centers**	**AP**
exemplar	artefact	actual point	actual point
parameter	K	K	s^* (penalty)
algorithm	greedy search	greedy search	message passing
performance	not stable	not stable	stable
complexity	$N \times K$	$N \times K$	$N^2 log(N)$

WHEN ? When averages don't make sense
WHY ? Stable, minimal distortion
CONS Computational complexity

Affinity Propagation

Given

$\mathcal{E} = \{e_1, e_2, ..., e_N\}$ *elements*

$d(e_i, e_j)$ *their dissimilarity*

Find $\sigma : \mathcal{E} \mapsto \mathcal{E}$ $\sigma(e_i)$, exemplar representing e_i

such that:

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

where $\begin{cases} S(e_i, e_j) = -d^2(e_i, e_j) & \text{if } i \neq j \\ S(e_i, e_i) = -s^* \end{cases}$ s^*: **penalty**

parameter

Particular cases

- $s^* = \infty$, only one exemplar 1 cluster
- $s^* = 0$, every point is an exemplar N clusters

Affinity Propagation, 2

Two types of messages

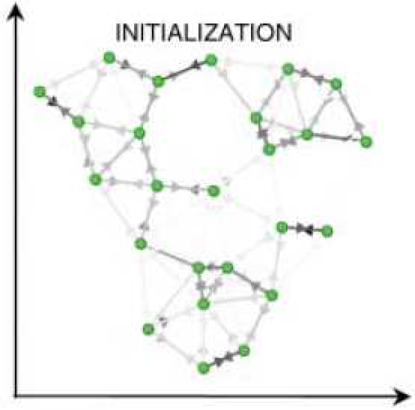
- $a(i,k)$: Availability of i as examplar for k
- $r(i,k)$: Responsibility of i to k

Rules of propagation

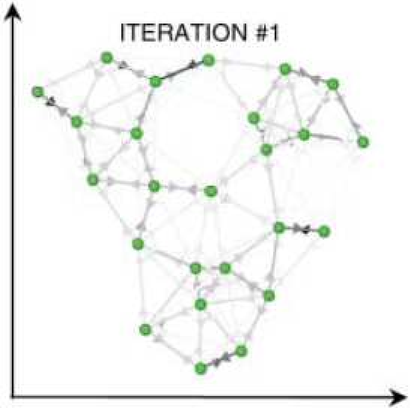
$$
\begin{aligned}
r(i,k) &= S(e_i,e_k) - \max_{k',k'\neq k}\{a(i,k') + S(e_i,e'_k)\} \\
r(k,k) &= S(e_k,e_k) - \max_{k',k'\neq k}\{S(e_k,e'_k)\}
\end{aligned}
$$

$$
\begin{aligned}
a(i,k) &= min\,\{0, r(k,k) + \textstyle\sum_{i',i'\neq i,k} \max\{0, r(i',k)\}\} \\
a(k,k) &= \textstyle\sum_{i',i'\neq k} \max\{0, r(i',k)\}
\end{aligned}
$$

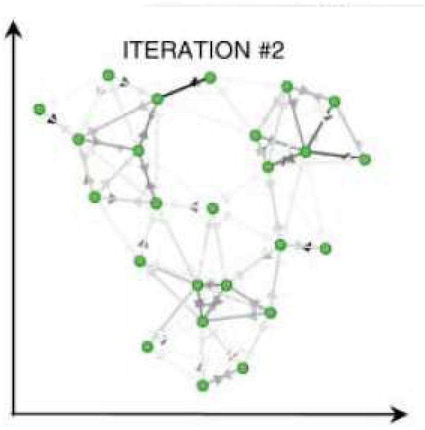
Iterations of Message passing



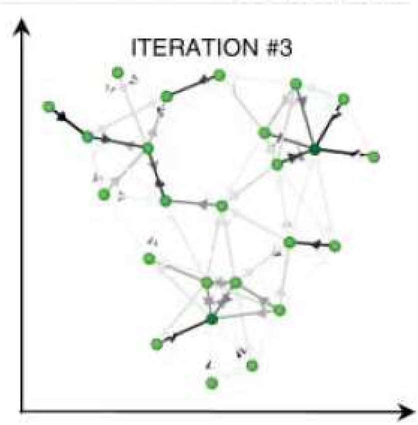
Iterations of Message passing



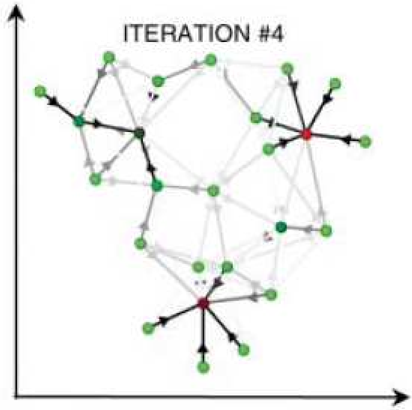
Iterations of Message passing



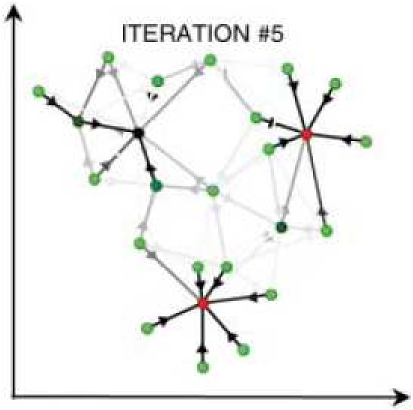
Iterations of Message passing



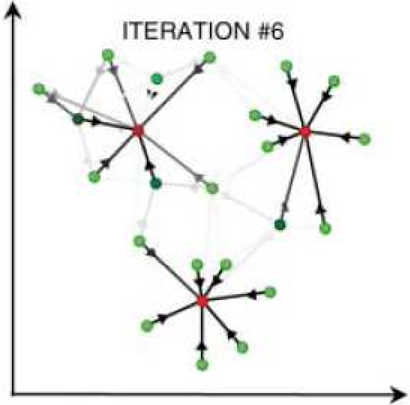
Iterations of Message passing



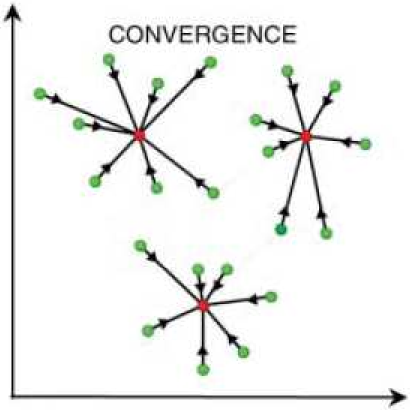
Iterations of Message passing



Iterations of Message passing

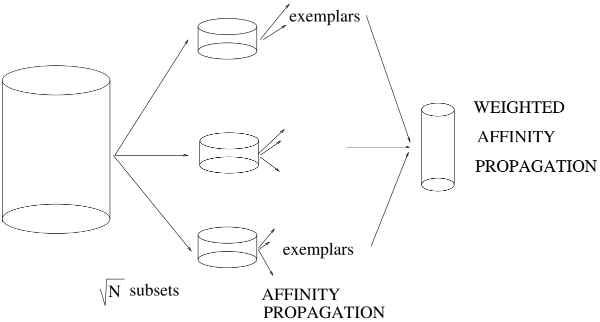


Iterations of Message passing



Hierarchical Affinity Propagation

ECML 2008



Thm

Let h be the height of the tree, b the branching factor, N_0 the size of each subproblem, K the average number of examplars for each sub problem. Then

$$C(h) \propto N^{\frac{h+2}{h+1}}$$

Extending AP to Data Streaming

StrAP : sketch

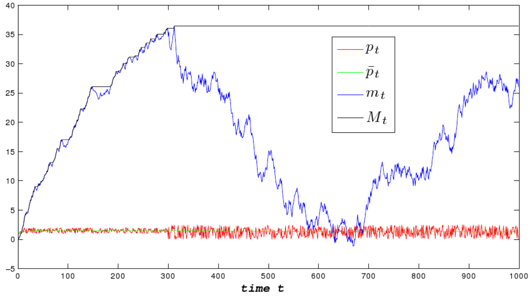
1. Job j_t arrives
2. Does it fit the current model $\mathcal{M}_t$?
   - YES: update $\mathcal{M}_t$
   - NO: $j_t \rightarrow$ Reservoir
3. Has the distribution changed ?
   - YES: build $\mathcal{M}_{t+1}$ from $\mathcal{M}_t$ and the reservoir

Stream Model: $\mathcal{M}_t = \{(j_i, n_i, \Sigma_i, t_i)\}$

- j_i examplar job
- n_i number of jobs represented by j_i
- Σ_i sum of distortions incurred by j_i
- t_i last time step when a job was affected to j_i

Has the distribution changed ?

Page-Hinkley statistical change detection test



$$
\begin{aligned}
\bar{p}_t &= \tfrac{1}{t}\textstyle\sum_{\ell=1}^{t} p_\ell \\
m_t &= \textstyle\sum_{\ell=1}^{t} \left(|p_\ell - \bar{p}_\ell| + \delta\right) \\
PH_t &= max\{m_\ell\} - m_t
\end{aligned}
$$

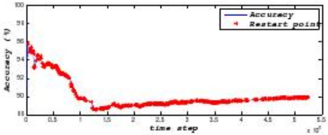
D. Hinkley. Inference about the change-point in a sequence of random variables. Biometrika, 1970

E. Page. Continuous inspection schemes. Biometrika, 1954

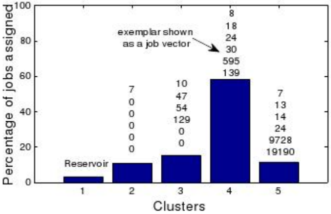
EGEE Job Streaming

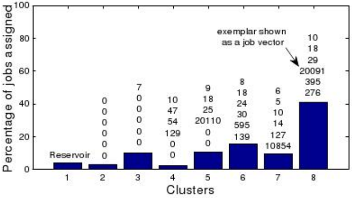
Dynamics of the distribution: schedule of restarts

Accuracy (succ/failed jobs)



Snapshots



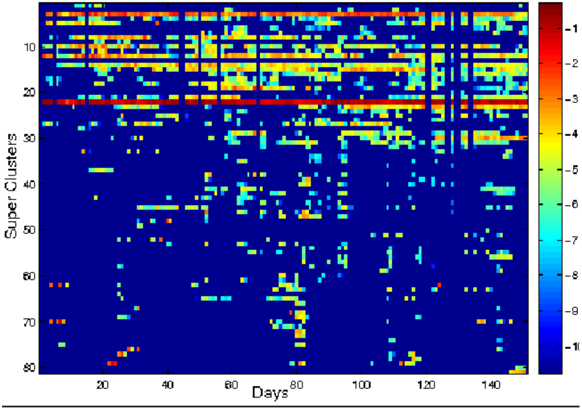


The EGEE traffic: months at a glance

A posteriori

build super-examplars from examplars — each s.e. a row

aggregate the traffic — along time



EGEE Job Streaming, end

Further work

1. List / Interpret outliers.
   Build a catalogue of situations
2. From job clustering to day clustering
   A day is a histogram of job clusters
3. Sequence modelling
   Caveat: nature of random variables
4. Fueling Job scheduling with realistic distribution models.

Overview