Optimal interval clustering: Application to Bregman clustering and statistical mixture learning

Frank Nielsen[1] Richard Nock[2]

[1]Sony Computer Science Laboratories/Ecole Polytechnique
[2]UAG-CEREGMIA/NICTA

May 2014

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

Hard clustering: Partitioning the data set

- Partition $\mathcal{X} = \{x_1, ..., x_n\} \subset \mathbb{X}$ into k clusters $\mathcal{C}_1 \subset \mathcal{X}, ..., \mathcal{C}_k \subset \mathcal{X}$:

$$\mathcal{X} = \biguplus_{i=1}^{k} \mathcal{C}_i$$

- Center-based (prototype) hard clustering: k-means [2], k-medians, k-center, ℓ_r-center [10], etc.
- Model-based hard clustering: statistical mixtures maximizing the complete likelihood (prototype=model paramter).

- k-means: NP-hard when $d > 1$ and $k > 1$ [11, 7, 1].
- k-medians and k-centers: NP-hard [12] (1984)
- In 1D, k-means is polynomial [3, 15]: $O(n^2 k)$.

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

Euclidean 1D k-means

- 1D k-means [8] has contiguous partition.
- Solved by enumerating all $\binom{n-1}{k-1}$ partitions in 1D (1958). Better than Stirling numbers of the second kind $S(n,k)$ that count all partitions.
- Polynomial in time $O(n^2k)$ using Dynamic Programming (DP) [3] (sketched in 1973 in two pages).
- R package `Ckmeans.1d.dp` [15] (2011).

Interval clustering: Structure

Sort $\mathcal{X} \in \mathbb{X}$ with respect to total order $<$ on $\mathbb{X}$ in $O(n \log n)$.

Output represented by:

- k intervals $I_i = [x_{l_i}, x_{r_i}]$ such that $\mathcal{C}_i = I_i \cap \mathcal{X}$.
- or better $k-1$ delimiters l_i ($i \in \{2, ..., k\}$) since $r_i = l_{i+1} - 1$ ($i < k$ and $r_k = n$) and $l_1 = 1$.

$$\underbrace{[x_1 ... x_{l_2 - 1}]}_{\mathcal{C}_1} \underbrace{[x_{l_2} ... x_{l_3 - 1}]}_{\mathcal{C}_2} ... \underbrace{[x_{l_k} ... x_n]}_{\mathcal{C}_k}$$

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

Objective function for interval clustering

Scalars $x_1 < ... < x_n$ are partitioned contiguously
into k clusters: $\mathcal{C}_1 < ... < \mathcal{C}_k$.

Clustering objective function:

$$\min e_k(\mathcal{X}) = \bigoplus_{j=1}^{k} e_1(\mathcal{C}_j)$$

$c_1(\cdot)$: intra-cluster cost/energy
$\oplus$: inter-cluster cost/energy (commutative, associative)

$n = kp + 1$ 1D points equally distributed $\rightarrow$ k different optimal clustering partitions

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

Examples of objective functions

In arbitrary dimension $\mathbb{X} = \mathbb{R}^d$:

- ℓ_r-clustering ($r \geq 1$): $\bigoplus = \sum$

$$e_1(\mathcal{C}_j) = \min_{p \in \mathbb{X}} \left(\sum_{x \in \mathcal{C}_j} d(x,p)^r \right)$$

  (argmin=prototype p_j is the same whether we take power of $\frac{1}{r}$ of sum or not)
  Euclidean ℓ_r-clustering: $r = 1$ median, $r = 2$ means.

- k-center ($\lim_{r \to \infty}$): $\bigoplus = \max$

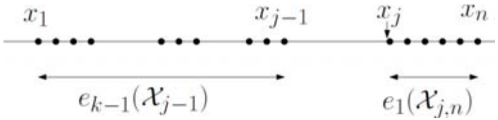
$$e_1(\mathcal{C}_i) = \min_{p \in \mathbb{X}} \max_{x \in \mathcal{C}_j} d(x,p)$$

- Discrete clustering: Search space in min is $\mathcal{C}_j$ instead of $\mathbb{X}$.

Note that in 1D, ℓ_s-norm distance is always $d(p,q) = |p - q|$, independent of $s \geq 1$.

Optimal interval clustering by Dynamic Programming

$\mathcal{X}_{j,i} = \{x_j, ..., x_i\}$ $(j \leq i)$
$\mathcal{X}_i = \mathcal{X}_{1,i} = \{x_1, ..., x_i\}$
$E = [e_{i,j}]$: $n \times k$ cost matrix, $O(n \times k)$ memory
$e_{i,m} = e_m(\mathcal{X}_i)$



Optimality equation:

$$e_{i,m} = \min_{m \leq j \leq i} \{e_{j-1,m-1} \oplus e_1(\mathcal{X}_{j,i})\}$$

Associative/commutative operator $\oplus$ ($+$ or max).
Initialize with $c_{i,1} = c_1(\mathcal{X}_i)$
E: compute from left to right column, from bottom to top.
Best clustering solution cost is at $e_{n,k}$.
Time: $n \times k \times O(n) \times T_1(n)$=$O(n^2 k T_1(n))$, O(nk) memory

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

Retrieving the solution: Backtracking

Use an auxiliary matrix $S = [s_{i,j}]$ for storing the argmin.

Backtrack in $O(k)$ time.

- Left index l_k of $\mathcal{C}_k$ stored at $s_{n,k}$: $l_k = s_{n,k}$.
- Iteratively retrieve the previous left interval indexes at entries $l_{j-1} = s_{l_j-1,i}$ for $j = k-1, ..., j = 1$.
  Note that $l_j - 1 = n - \sum_{l=j}^{k} n_l$ and $l_j - 1 = \sum_{l=1}^{j-1} n_l$.

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

Optimizing time with a Look Up Table (LUT)

Save time when computing $e_1(\mathcal{X}_{j,i})$ since we perform $n \times k \times O(n)$ such computations.

Look Up Table (LUT): Add extra $n \times n$ matrix E_1 with $E_1[j][i] = e_1(\mathcal{X}_{j,i})$.
Build in $O(n^2 T_1(n))$...
Then DP in $O(n^2 k)$=$O(n^2 T_1(n)))$.

$\rightarrow$ quadratic amount of memory ($n > 10000$...)

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

DP solver with cluster size constraints

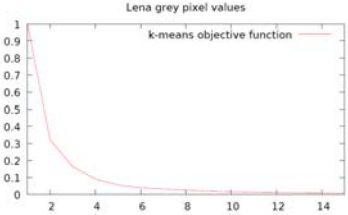
n_i^- and n_i^+: lower/upper bound constraints on $n_i = |\mathcal{C}_i|$
$\sum_{l=1}^{k} = n_i^- \leq n$ and $\sum_{l=1}^{k} = n_i^+ \geq n$.
When no constraints: add dummy constraints $n_i^- = 1$ and $n_i^+ = n - k + 1$.

$n_m = |\mathcal{C}_m| = i - j + 1$ such that $n_m^- \leq n_m \leq n_m^+$.
$\rightarrow j \leq i + 1 - n_m^-$ and $j \geq i + 1 - n_m^+$.

$$e_{i,m} = \min_{\substack{\max\{1+\sum_{l=1}^{m-1} n_l^-, i+1-n_m^+\} \leq j \\ j \leq i+1-n_m^-}} \left\{ e_{j-1,m-1} \oplus e_1(\mathcal{X}_{j,i}) \right\},$$

Model selection from the DP table

$m(k) = \frac{e_k(\mathcal{X})}{e_1(\mathcal{X})}$ decreases with k and reaches minimum when $k = n$.
Model selection: trade-off choose *best model* among all the models with $k \in [1, n]$.



Regularized objective function: $e'_k(\mathcal{X}) = e_k(\mathcal{X}) + f(k)$, $f(k)$ related to model complexity.
Compute the DP table for $k = n, ..., 1$ and avoids redundant computations.
Then compute the criterion for the last line (indexed by n) and choose the argmin of e'_k.

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

A Voronoi cell condition for DP optimality

elements $\rightarrow$ interval clusters $\rightarrow$ prototypes

interval clusters $\leftarrow$ prototypes

Voronoi diagram:
Partition $\mathbb{X}$ wrt. $\mathcal{P} = \{p_1, ..., p_k\}$.
Voronoi cell:

$$V(p_j) = \{x \in \mathbb{X} \ : \ d^r(x, p_j) \leq d^r(x, p_l) \ \forall l \in \{1, ..., k\}\}.$$

x^r is a monotonically increasing function on $\mathbb{R}^+$, equivalent to

$$V'(p_j) = \{x \in \mathbb{X} \ : \ d(x : p_j) < d(x : p_l)\}$$

DP guarantees optimal clustering when $\forall \mathcal{P}$, $V'(p_j)$ is an interval
2-clustering exhibits the Voronoi bisector.

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

1-mean (centroid): $O(n)$ time

$$\min_p \sum_{i=1}^{n} (x_i - p)^2$$

$$D(x,p) = (x-p)^2, \quad D'(x,p) = 2(x-p), \quad D''(x,p) = 2$$

Convex optimization (existence and unique solution)

$$\sum_{i=1}^{n} D'(x,p) = 0 \Rightarrow \sum_{i=1}^{n} x_i - np = 0$$

Center of mass $p = \frac{1}{n}\sum_{i=1}^{n} x_i -$ (barycenter)
Extends to Bregman divergence:

$$D_F(x,p) = F(x) - F(p) - (x-p)F'(p)$$

2-means: $O(n \log n)$ time

Find x_{l_2} ($n-1$ potential locations for x_l: from x_2 to x_n):

$$\min_{x_{l_2}} \{e_1(\mathcal{C}_1) + e_1(\mathcal{C}_2)\}$$

Browse from left to right $l_2 = x_2, ..., x_n$.
Update cost in constant time $E_2(l+1)$ from $E_2(l)$ (SATs also $O(1)$):

$$E_2(l) = e_2(x_1 ... x_{l-1} | x_l ... x_n)$$

$$\mu_1(l+1) = \frac{(l-1)\mu_1(l) + x_l}{l}, \quad \mu_2(l+1) = \frac{(n-l+1)\mu_2(l) - x_l}{n-l}$$

$$v_1(l+1) = \sum_{i=1}^{l} (x_i - \mu_1(l+1))^2 = \sum_{i=1}^{l} x_i^2 - l\mu_1^2(l+1)$$

$$\Delta E_2(l) = \frac{l-1}{l} \|\mu_1(l) - x_l\|^2 + \frac{n-l+1}{n-l} \|\mu_2(l) - x_l\|^2$$

2-means: Experiments

Intel Win7 i7-4800

n	Brute force	SAT	Incremental
300000	155.022	0.010	0.0091
1000000	1814.44	0.018	0.015

Do we need sorting and $\Omega(n \log n)$ time? ($k = 1$ is linear time)

Note that MaxGap does not yield the separator (because centroid is sum of squared distance minimizer)

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

Optimal 1D Bregman k-means

Bregman information [2] e_1 (generalizes cluster variance):

$$e_1(\mathcal{C}_j) = \min_{x_l \in \mathcal{C}_j} w_l B_F(x_l : p_j). \tag{1}$$

Expressed as [14]:
$e_1(\mathcal{C}_j) = \left(\sum_{x_l \in \mathcal{C}_j} w_l\right)(p_j F'(p_j) - F(p_j)) + \left(\sum_{x_l \in \mathcal{C}_j} w_l F(x_l)\right) - F'(p_j)\left(\sum_{x \in \mathcal{C}_j} w_l x\right)$
process using *Summed Area Tables* [6] (SATs)
$S_1(j) = \sum_{l=1}^{j} w_l$, $S_2(j) = \sum_{l=1}^{j} w_l x_l$, and $S_3(j) = \sum_{l=1}^{j} w_l F(x_l)$ in $O(n)$ time at preprocessing stage.
Evaluate the Bregman information $e_1(\mathcal{X}_{j,i})$ in constant time $O(1)$.
For example, $\sum_{l=j}^{i} w_l F(x_l) = S_3(i) - S_3(j-1)$ with $S_3(0) = 0$.
Bregman Voronoi diagrams have connected cells [4] thus DP yields optimal interval clustering.

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

Exponential families in statistics

Family of probability distributions:

$$\mathcal{F} = \{p_F(x;\theta) \ : \ \theta \in \Theta\}$$

Exponential families [13]:

$$p_F(x|\theta) = \exp\left(t(x)\theta - F(\theta) + k(x)\right),$$

For example:
univariate Rayleigh $R(\sigma)$, $t(x) = x^2$, $k(x) = \log x$, $\theta = -\frac{1}{2\sigma^2}$, $\eta = -\frac{1}{\theta}$, $F(\theta) = \log -\frac{1}{2\theta}$ and $F^*(\eta) = -1 + \log\frac{2}{\eta}$.

Uniorder exponential families: MLE

Maximum Likelihood Estimator (MLE) [13]:

$$e_1(\mathcal{X}_{j,i}) = \hat{l}(x_j, ..., x_i) = F^*(\hat{\eta}_{j,i}) + \frac{1}{i-j+1}\sum_{l=j}^{i} k(x_l).$$

with $\hat{\eta}_{j,i} = \frac{1}{i-j+1}\sum_{l=j}^{i} t(x_l)$.
By making a change of variable $y_l = t(x_l)$, and not accounting the $\sum k(x_l)$ terms that are constant for any clustering, we get

$$e_1(\mathcal{X}_{j,i}) \equiv F^*\left(\frac{1}{i-j+1}\sum_{l=j}^{i} y_l\right)$$

Hard clustering for learning statistical mixtures

Expectation-Maximization learns monotonically from an initialization by maximizing the incomplete log-likelihood.
Mixture maximizing the complete log-likelihood:

$$l_c(\mathcal{X}; L, \Omega) = \sum_{i=1}^{n} \log(\alpha_{l_i} p(x_i; \theta_{l_i})),$$

$L = \{l_i\}_i$: hidden labels.

$$\max l_c \equiv \min_{\theta_1,\ldots,\theta_k} \sum_{i=1}^{n} \min_{j=1}^{k} (-\log p(x_i; \theta_j) - \log \alpha_j).$$

Given fixed α and $-\log p_F(x; \theta)$ amounts to a dual Bregman divergence[2].
Run Bregman k-means and DP yields optimal partition since additively-weighted Bregman Voronoi diagrams are interval [4].

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

Hard clustering for learning statistical mixtures

Location families:

$$\mathcal{F} = \{f(x;\mu) = \frac{1}{\sigma} f_0(\frac{x-\mu}{\sigma}), \mu \in \mathbb{R}\}$$

f_0 standard density, $\sigma > 0$ fixed. Cauchy or Laplacian families have density graphs intersecting in exactly one point.

$\rightarrow$ singly-connected Maximum Likelihood Voronoi cells.

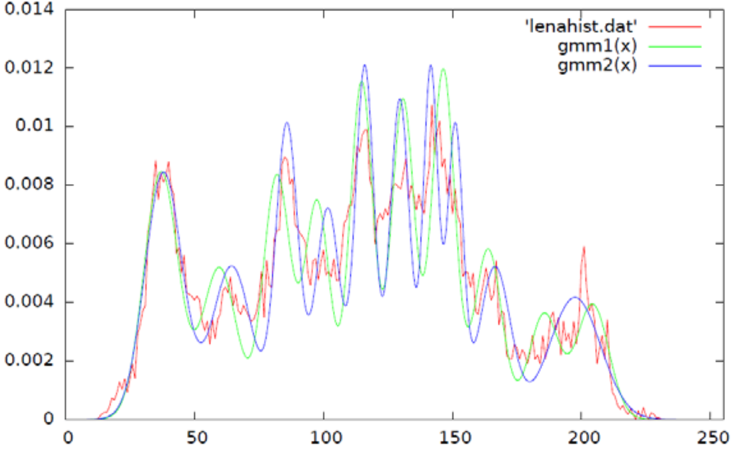
Model selection: Akaike Information Criterion [5] (AIC):

$$\mathrm{AIC}(x_1, ..., x_n) = -2l(x_1, ..., x_n) + 2k + \frac{2k(k+1)}{n-k-1}$$

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

Experiments with: Gaussian Mixture Models (GMMs)

gmm_1 score $= -3.0754314021966658$ (Euclidean k-means) gmm_2 score $= -3.038795325884112$ (Bregman k-means, better)



© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

Conclusion

- Generic DP for solving interval clustering:
  - $O(n^2 k T_1(n))$-time using $O(nk)$ memory
  - $O(n^2 T_1(n))$ time using $O(n^2)$ memory
- Refine DP by adding minimum/maximum cluster size constraints
- Model selection from DP table
- Two applications:
  - 1D Bregman ℓ_r-clustering. 1D Bregman k-means in $O(n^2 k)$ time using $O(nk)$ memory using Summed Area Tables (SATs)
  - Mixture learning maximizing the complete likelihood:
    - For uni-order exponential families amount to a dual Bregman k-means on $\mathcal{Y} = \{y_i = t(x_i)\}_i$
    - For location families with density graph intersecting pairwise in one point (Cauchy, Laplacian: $\notin$ exponential families)

Perspectives

$\Omega(n \log n)$ for sorting.

Hierarchical center-based clustering with single-linkage: clustering tree.
Best k-partition pruning using DP [**?**]:
Optimal for $\alpha = 2 + \sqrt{3}$-perturbation resilient instances.
Time $O(nk^2 + nT_1(n))$
Question: How to maintain dynamically an optimal contiguous clustering? (core-set approximation in the streaming model [9])

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

Bibliography I

Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat.
Np-hardness of Euclidean sum-of-squares clustering.
Machine Learning, 75(2):245–248, 2009.

Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh.
Clustering with Bregman divergences.
Journal of Machine Learning Research, 6:1705–1749, 2005.

Richard Bellman.
A note on cluster analysis and dynamic programming.
Mathematical Biosciences, 18(3-4):311 – 312, 1973.

Jean-Daniel Boissonnat, Frank Nielsen, and Richard Nock.
Bregman Voronoi diagrams.
Discrete Computational Geometry, 44(2):281–307, September 2010.

J. Cavanaugh.
Unifying the derivations for the Akaike and corrected Akaike information criteria.
Statistics & Probability Letters, 33(2):201–208, April 1997.

Franklin C. Crow.
Summed-area tables for texture mapping.
In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, pages 207–212, New York, NY, USA, 1984. ACM.

Sanjoy Dasgupta.
The hardness of k-means clustering.
Technical Report CS2008-0916.

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

Bibliography II

Walter D Fisher.
On grouping for maximum homogeneity.
Journal of the American Statistical Association, 53(284):789–798, 1958.

Sariel Har-Peled and Akash Kushal.
Smaller coresets for k-median and k-means clustering.
In *Proceedings of the Twenty-first Annual Symposium on Computational Geometry*, SCG '05, pages 126–134, New York, NY, USA, 2005. ACM.

Meizhu Liu, Baba C. Vemuri, Shun ichi Amari, and Frank Nielsen.
Shape retrieval using hierarchical total Bregman soft clustering.
IEEE Trans. Pattern Anal. Mach. Intell., 34(12):2407–2419, 2012.

Meena Mahajan, Prajakta Nimbhorkar, and Kasturi R. Varadarajan.
The planar k-means problem is NP-hard.
Theoretical Computer Science, 442:13–21, 2012.

Nimrod Megiddo and Kenneth J Supowit.
On the complexity of some common geometric location problems.
SIAM journal on computing, 13(1):182–196, 1984.

Frank Nielsen.
k-mle: A fast algorithm for learning statistical mixture models.
CoRR, abs/1203.5181, 2012.

Frank Nielsen and Richard Nock.
Sided and symmetrized Bregman centroids.
IEEE Transactions on Information Theory, 55(6):2882–2904, 2009.

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.

Bibliography III

Haizhou Wang and Mingzhou Song.
Ckmeans.1d.dp: Optimal k-means clustering in one dimension by dynamic programming.
R Journal, 3(2), 2011.

© 2014 Frank Nielsen, Sony Computer Science Laboratories, Inc.