Information-theoretic clustering with applications

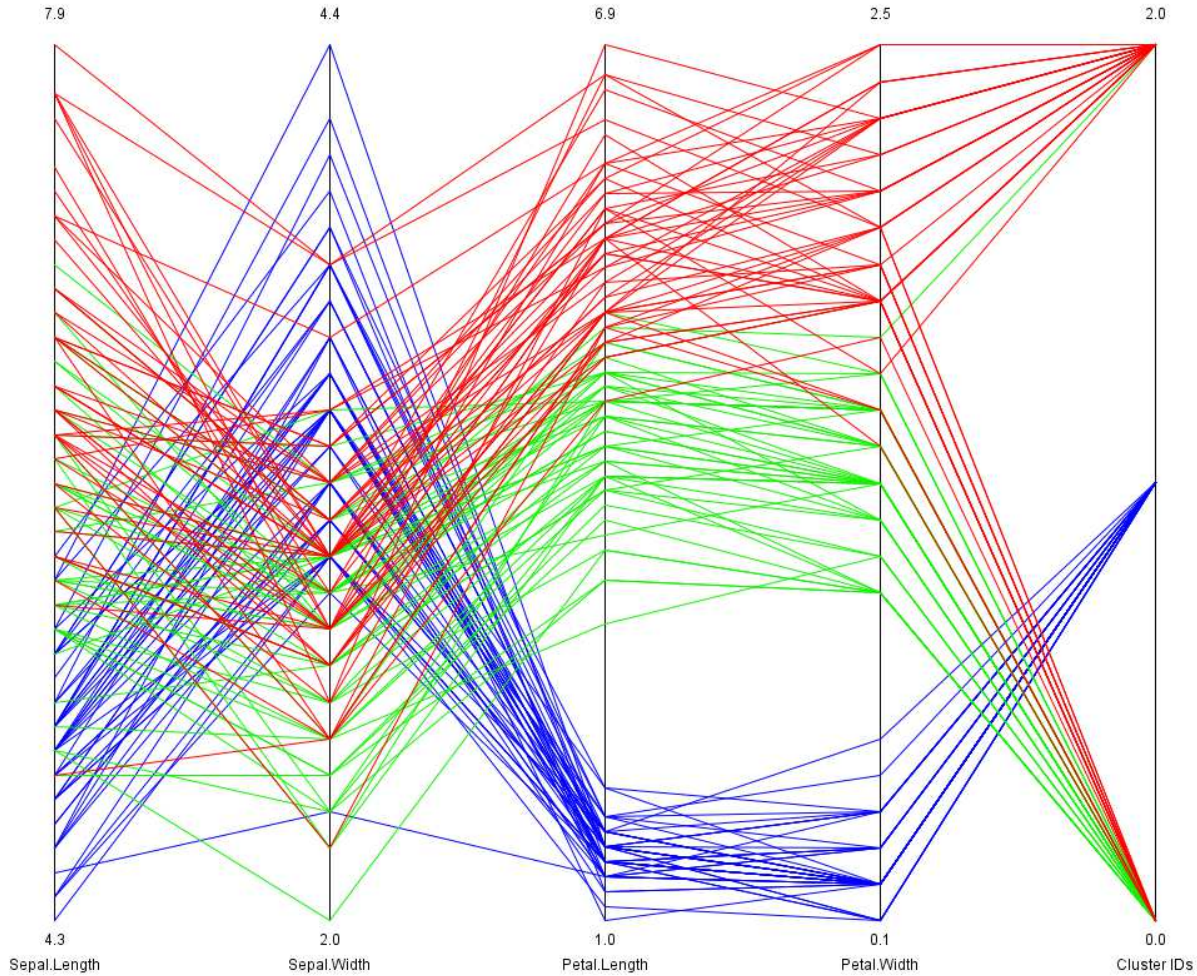
Frank Nielsen

Sony Computer Science Laboratories Inc
École Polytechnique

Kyoto University Information Seminar
9th June 2014

Clustering: Data exploratory science

- Clustering: Find homogeneous groups in data
- Clustering: Separate data into groups



`Iris` data set from UCI: $d = 4$, $k = 3$, $n = 150$

Clustering: What to see?, when to see ?

- Distance or similarity measure between data?
- Data membership: Hard or soft clustering?
- How many groups?
- Outliers?
- "Shapes" of groups?
- Representation of data and cleaning,
- Missing data, truncated data, ...
- Etc.

Outline

Clustering n data in d dimensions into k clusters
Usual case: $d << k << n$, but all cases possible

- Quick tutorial: "The essentials"
  - Ordinary$k$-means,
  - Gaussian mixtures models,
  - Model selection,
  - Other kinds of clustering, etc.
- Optimal 1D contiguous clustering
- Clustering histograms with Jeffreys divergence

Part I

Quick tutorial

k-means clustering

Partition the data set $\mathcal{X} = \{x_1, ..., x_n\}$ into k groups

$$\mathcal{P} = \{\mathcal{G}_1, ..., \mathcal{G}_k\},$$

each group $\mathcal{G}_j$ has a center c_j.

Minimize the objective function (cost, energy, loss):

$$e(\mathcal{X}, \mathcal{C}) = \sum_{i=1}^{n} \min_{j=1}^{k} \|x_i - c_j\|^2$$

- NP-hard when $d > 1$ and $k > 1$
- For $d = 1$, center c_1 is the centroid and $e_1(\mathcal{X}) = e(\mathcal{X}, \{\mathcal{C}_1\}) = \mathrm{var}(\mathcal{X})$, the "unnormalized" variance: $\mathrm{var}(\mathcal{X}) = \sum_i \|x_i\|^2 - n\|c_1\|^2$

Rewriting the k-means cost to minimize

$$
\begin{aligned}
e(\mathcal{X},\mathcal{C}) &= \sum_{i=1}^{n} \min_{j=1}^{k} \|x_i - c_j\|^2, \\
&= \frac{1}{2}\sum_{j=1}^{k} \sum_{x,x' \in \mathcal{G}_j} \|x - x'\|^2 + \text{constant}, \\
&= -\frac{1}{2}\sum_{j=1}^{k} \sum_{x \in \mathcal{G}_j} \sum_{x' \notin \mathcal{G}_k} \|x - x'\|^2 + \text{constant}, \\
&= \sum_{j=1}^{k} n_j \mathrm{var}(\mathcal{G}_j), \text{size cluster } \mathcal{G}_i\text{: } n_j = |\mathcal{G}_i| \\
&= \mathrm{var}(\mathcal{X}) - \mathrm{var}(\mathcal{Y}), \text{where } \mathcal{Y} = \{(n_j, c_j)\}_{j=1}^{k}
\end{aligned}
$$

Interpreted as: min sum cluster inter-distances, max sum cluster inter-distances, min sum of cluster variances, min quantization loss.

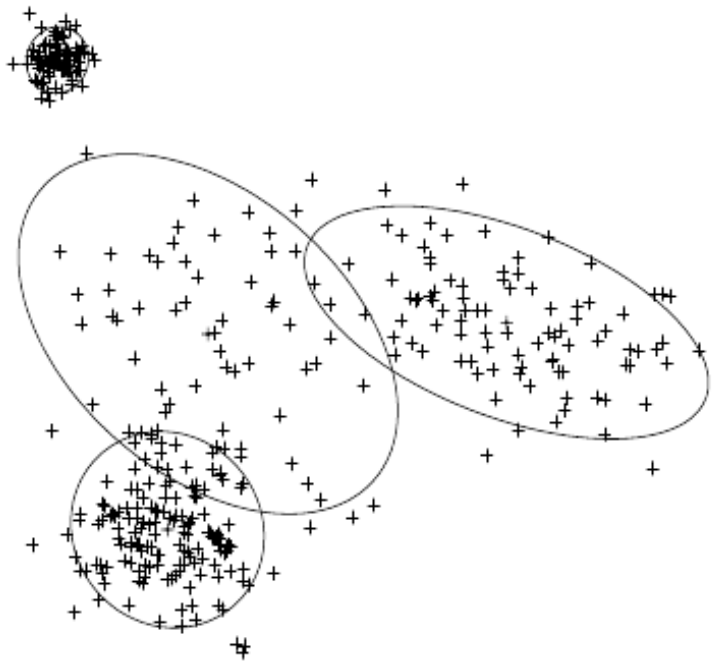
Heuristics for k-means

- Local heuristics:
  - Lloyd (batched): assign data to closest center, remap cluster centers to centroids, reiterate until convergence
  - McQueen (incremental, online): add a point at a time, assign x to closest cluster, update that cluster centroid, reiterate until convergence
  - Hartigan (single-point): find a point $x \in \mathcal{G}_i$ and a cluster $\mathcal{G}_j$ so that relocating $x \in \mathcal{G}_j$ decreases k-means cost, reiterate until convergence
- Global heuristics:
  - Forgy: random seeding. Best Forgy is 2-approximation via the variance-bias decomposition:
    $D(x, \mathcal{X}) = \sum_{i=1}^{n} \|x - x_i\|^2 = \mathrm{var}(\mathcal{X}) + n\|x - c_1\|^2$.
  - Fastest fast traversal: choose c_1 at random, then c_i as the point x farthest from $\{c_1, ..., c_{i-1}\}$, repeat.
  - Randomized k-means++ (probabilistic $\tilde{O}(\log k)$ bound)
  - Global k-means

Probabilistic model-based clustering

Maximize likelihood for the mixture density:

$$m(x) = \sum_{i=1}^{k} w_i p(x; \mu_i, \Sigma_i)$$



Statistical mixture models: generative models

Statistical Gaussian Mixtures Models

Universal smooth density estimator.

Gaussian distribution:

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|\Sigma|}} e^{-\frac{1}{2}D_{\Sigma^{-1}}(x-\mu,x-\mu)}$$

Related to the squared Mahalanobis distance:

$$D_Q(x,y) = (x-y)^T Q(x-y), x \in \mathbb{R}^d$$

$\rightarrow$ log-concave density

Expectation-maximization (EM) algorithm: maximize incomplete likelihood.

EM tends to k-means when $\Sigma_j = \lambda I$ with $\lambda \rightarrow 0$ (or any fixed λ, see [34]).

Sampling from a Gaussian Mixture Model

To sample a variate x from a GMM:

- Choose a component l according to the weight distribution $w_1, ..., w_k$,
- Draw a variate x according to $N(\mu_l, \Sigma_l)$.

$\rightarrow$ Sampling is a doubly stochastic process:

- throw a biased dice with k faces to choose the component:

$$l \sim \text{Multinomial}(w_1, ..., w_k)$$

(normalized histogram.)

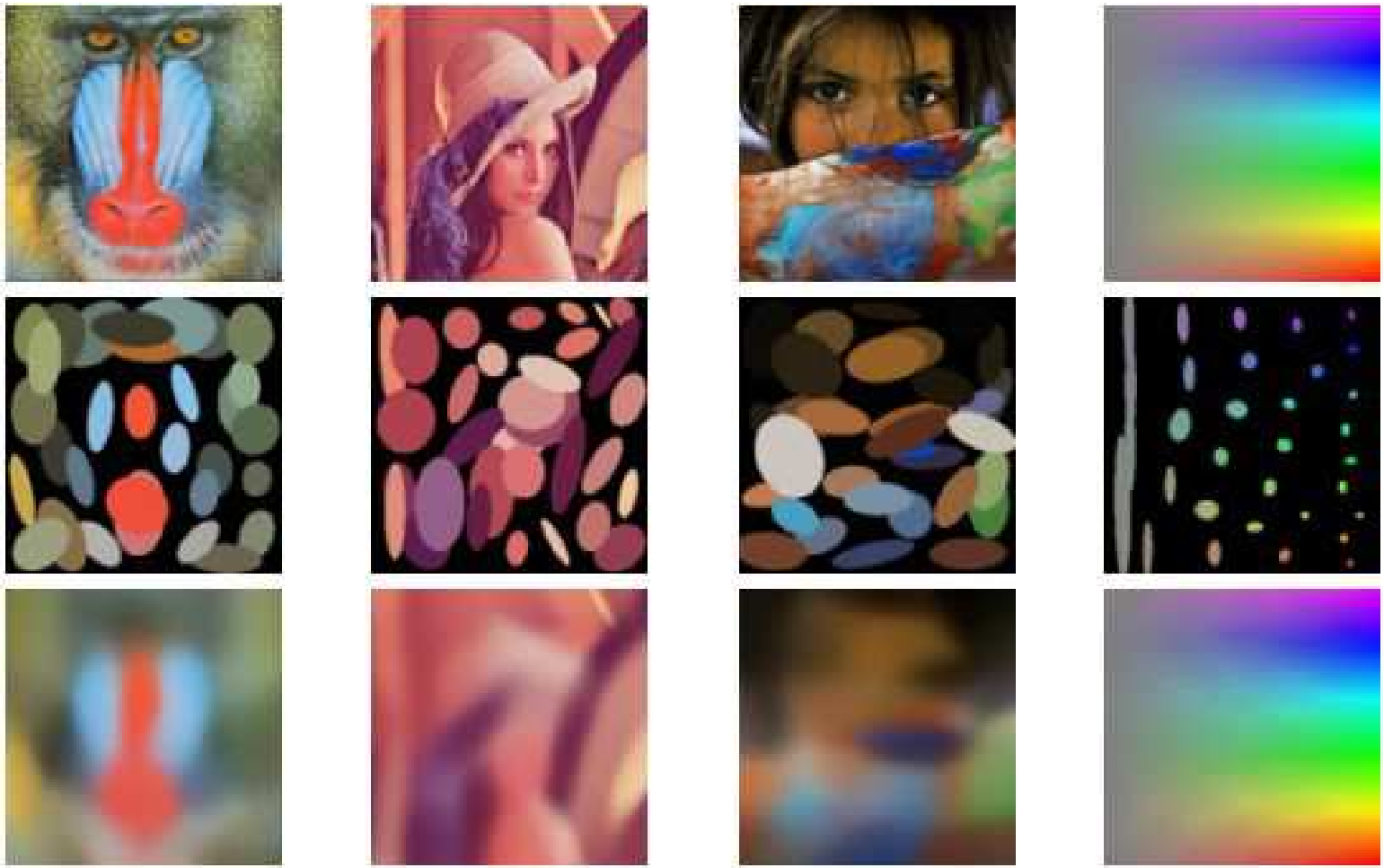
- then draw at random a variate x from the l-th component

$$x \sim \text{Normal}(\mu_l, \Sigma_l)$$

$x = \mu + Cz$ with Cholesky: $\Sigma = CC^\top$ and $z = [z_1 \ ... \ z_d]^\top$
standard normal random variate: $z_i = \sqrt{-2 \log U_1} \cos(2\pi U_2)$

GMMs: Generative models, sampling [15]

A pixel (x,y,R,G,B) : A data point in 5D



Model selection: Choosing k

The more parameters the better the model but it over fits.

Solution: Penalized likelihood to maximize.

Bayesian Information Criterion (BIC):

$$\max l(\theta) - \frac{\#\theta}{2} \log n$$

Clustering analysis

- Parametric clustering (k given, cost-based)
  - center-based hard clustering (k-means, k-medians, min diameter, single linkage, etc.)
  - model-based soft clustering
- Non-parametric clustering (k adjusts with data)
  - Kernel density estimators (mean shift)
  - Dirichlet process mixture models
- Hierarchical clustering
- Graph-based clustering (normalized cut)
- Affinity propagation
- Subspace/manifold clustering
- Etc!

Cluster validation (distance between clustering, confusion matrix, NMI, Rand index, etc.)

Part II

Optimal 1D contiguous clustering

Hard clustering: Partitioning the data set

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

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

- Center-based hard clustering (center=prototype): k-means [1], k-medians, k-center, ℓ_r-center [21], etc.
- Model-based hard clustering: statistical mixtures maximizing the complete likelihood (prototype=model parameter).

k-means clustering

Minimize the sum of intra-cluster variances:

$$\min_{p_1,\ldots,p_k} \sum_{i=1}^{n} \min_{j=1}^{k} \|x_i - p_j\|^2$$

- k-means: NP-hard when $d > 1$ and $k > 1$ [24, 12]. k-medians and k-centers also NP-hard [25] (1984)
- Global heuristics (for initialization) (Forgy [14], global k-means [40], k-means++ [4], etc.) and local search heuristics (incremental Voronoi MacQueen [23], batched Voronoi Lloyd [22], single-point swap Hartigan [17], etc.)
- In 1D, k-means is polynomial [3, 39]: $O(n^2 k)$.

Euclidean 1D k-means

- 1D k-means [13] 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` [39] (2011).

(Sketch of the) DP solution [3] (Bellman, 1973)

MATHEMATICAL BIOSCIENCES **18**, 311–312 (1973)

A Note on Cluster Analysis and Dynamic Programming*

RICHARD BELLMAN

Departments of Mathematics, Electrical Engineering, and Medicine, University of Southern California, Los Angeles, California 90007

ABSTRACT

In a number of situations a set of points falls naturally into clumps or clusters. It is often easy to recognize these groups visually, but not as easy to use a computer to perform the same action when the input is a set of numbers. The application of dynamic programming to this pattern recognition task in the one-dimensional case is considered.

1. INTRODUCTION

In a number of situations a set of points falls naturally into clumps or clusters. It is often easy to recognize these groups visually, but not as easy to use a computer to perform the same action when the input is a set of numbers. The purpose of this note is to indicate the application of dynamic programming to this pattern recognition task in the one-dimensional case. Subsequently we shall examine multidimensional versions.

2. CLUSTER ANALYSIS

Consider a line segment with the points 0, 1, 2, . . ., N. Some of these points are occupied by X's, some unoccupied, as shown below.

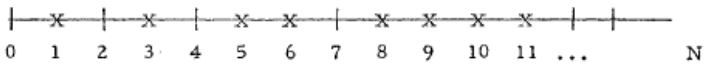


FIG 1.

We recognize two clusters immediately. Suppose, however, the foregoing information is given digitally. We are told that the set of occupied points is [1, 3, 5, 6, 8, 9, 10, 11, . . .], or alternatively given the set of unoccupied

* Supported by National Institutes of Health under Grant No. GM 16197-03, Caltech President's Fund Grant No. PF 019, and NASA Contract No. NAS 7-100.

sites, and asked to provide an algorithm which will allow the grouping to be done by a digital computer.

To begin with, we must make precise what we mean by a "cluster." It is necessary then to define a numerical measure of homogeneity. There are a number of different features we can employ: size of interval, number of occupied points in the interval, minimum distance between neighboring occupied sites, maximum distance between neighboring occupied sites, and so on. Taking these factors into account, we can assign a measure of homogeneity, $m(k_1, k_2)$, to the set of points contained in any interval $[k_1, k_2]$ where $0 \leqslant k_1 \leqslant k_2 \leqslant N$, with k_1 and k_2 integers. We do not dwell on this aspect of the problem because the method given below is independent of the form of this measure.

If we then choose intervals $[k_1, k_2]$, $[k_3, k_4]$, . . . to contain our choice of clusters, we see that the task of choosing clusters in an optimal fashion is equivalent to that of maximizing the sum

$$m(k_1, k_2) + m(k_3, k_4) + \cdots, \tag{1}$$

where $0 \leqslant k_1 < k_2 < k_3 < k_4 \cdots \leqslant N$.

There are actually two distinct types of problems here. The first is: Divide the set of points in $[0, N]$ into M clusters in an optimal fashion, where M is an assigned integer. The second is: Determine the optimal value of M, and then the optimal subdivision.

We shall solve the second problem using the solution of the first.

3. APPLICATION OF DYNAMIC PROGRAMMING

We begin with the observation that we can regard the choice of clusters, i.e. the intervals $[k_{2i-1}, k_{2i}]$ as a multistage process in which we choose first the Mth interval $[k_{2M-1}, k_{2M}]$, then the $(M-1)$st interval, and so on.

Let us then introduce the function

$$f_M(N) = \max_{\{k_i\}} [m(k_1, k_2) + \cdots + m(k_{2M-3}, k_{2M-2}) + m(k_{2M-1}, k_{2M})]. \tag{2}$$

The principle of optimality then yields the functional equation

$$f_M(N) = \max_{0 \leqslant k_{2M-1} \leqslant k_{2M} \leqslant N} [m(k_{2M-1}, k_{2M}) + f_{M-1}(k_{2M-1})], \tag{3}$$

$M = 1, 2, \ldots$. We set $f_0(N) = 0$, $f_M(0) = 0$, $M = 0, 1, \ldots$.

The numerical solution of an equation of this nature is a simple matter with a contemporary computer for N of the order of 10^6. Once $f_M(N)$ has been calculated for a range of M-values, we calculate

$$\max_M f_M(N). \tag{4}$$

This yields the solution of the second problem.

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 $l_1 = 1$.

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

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)

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

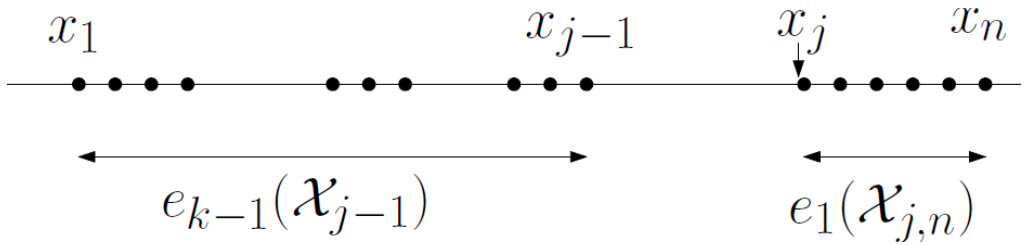
$$e_1(\mathcal{C}_i) = \min_{p \in \mathbb{X}} \max_{x \in \mathcal{C}_j} d(x,p)$$

- 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

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 $l_j - 1 = \sum_{l=1}^{j-1} n_l$.

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 ($n > 10000$...)

DP solver with cluster size constraints

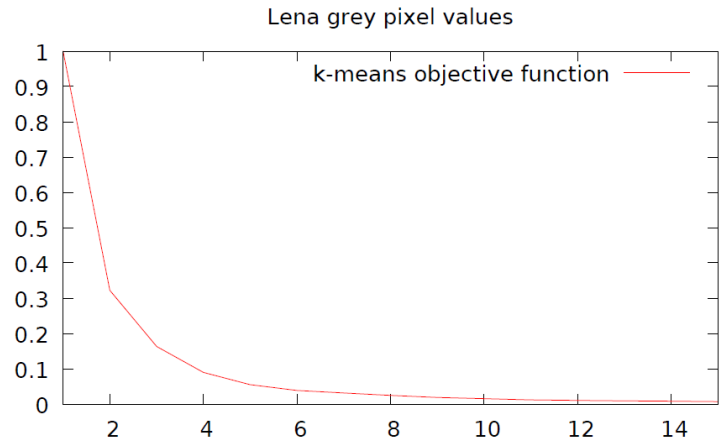
n_i^- and n_i^+: lower/upper bound constraints on $n_i = |\mathcal{C}_i|$
$\sum_{l=1}^{k} = n_i^- \leq n$ and $\sum_{l=1}^{k} = n_i^+ \geq n$.
When no constraints: add dummy constraints $n_i^- = 1$ and $n_i^+ = n - k - 1$.

$n_m = |\mathcal{C}_m| = i - j + 1$ 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.

A Voronoi cell condition for DP optimality

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

interval clusters $\leftarrow$ prototypes

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

Optimal 1D Bregman k-means

Bregman information [1] e_1 (generalizes cluster variance):

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

Expressed as [35]:
$e_1(\mathcal{C}_j) = \left(\sum_{x_l \in \mathcal{C}_j} w_l\right)(p_j F'(p_j) - F(p_j)) + \left(\sum_{x_l \in \mathcal{C}_j} w_l F(x_l)\right) - F'(p_j)\left(\sum_{x \in \mathcal{C}_j} w_l x\right)$
process using *Summed Area Tables* [10] (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 [6] thus DP yields optimal interval clustering.

Exponential families in statistics

Family of probability distributions:

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

Exponential families [30]:

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

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) [30]:

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

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[1].
Run Bregman k-means and DP yields optimal partition since additively-weighted Bregman Voronoi diagrams are interval [6].

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.

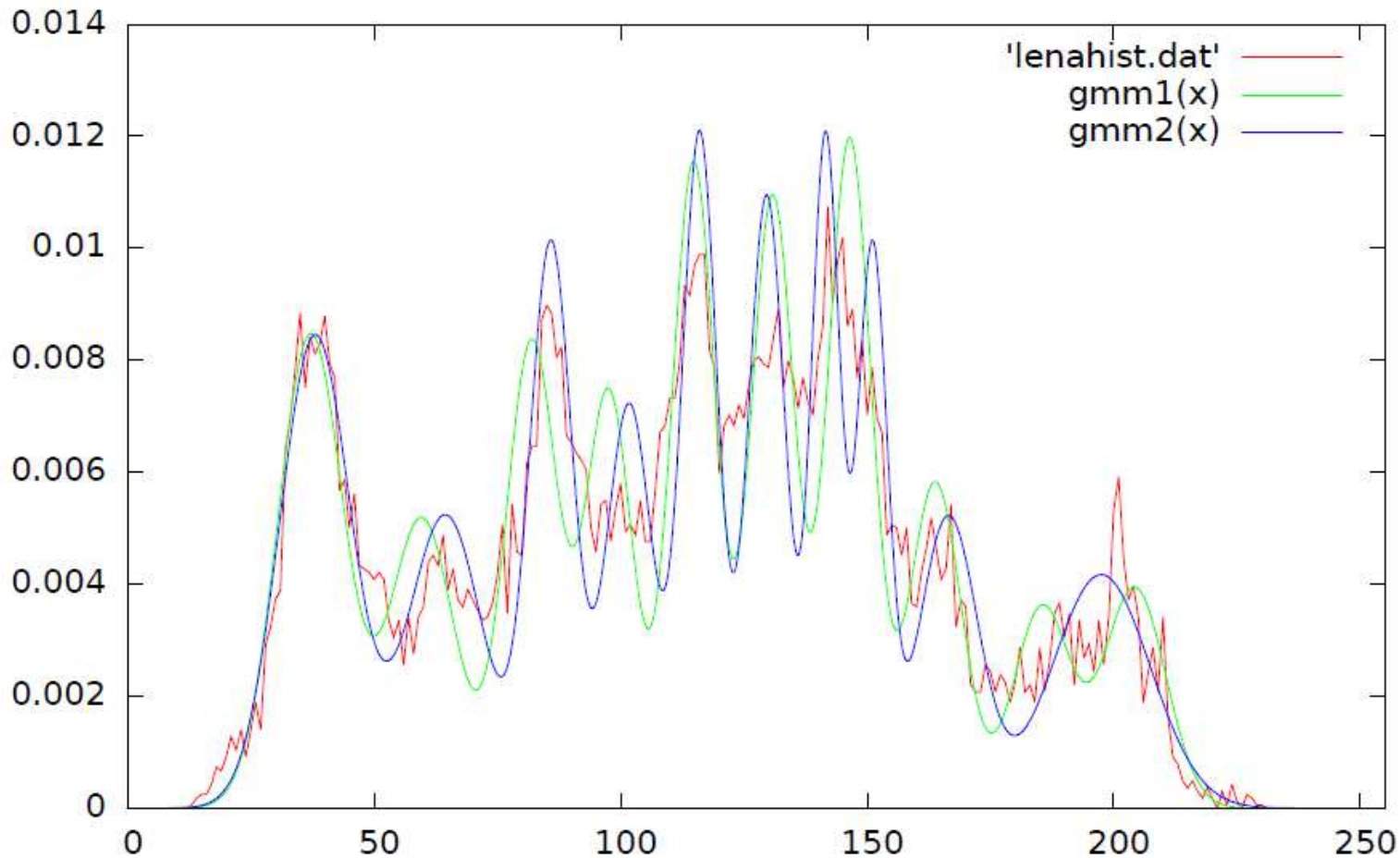
$\rightarrow$ singly-connected Maximum Likelihood Voronoi cells.

Model selection: Akaike Information Criterion [7] (AIC):

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

Experiments with: Gaussian Mixture Models (GMMs)

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



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)

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)

Part III

Clustering histograms with Jeffreys divergence

Why histogram clustering?

Task: Classify documents into categories:
Bag-of-Word (BoW) modeling paradigm [5, 11].

- Define a word dictionary, and
- Represent each document by a *word count* histogram.

Centroid-based k-means clustering [1]:

- Cluster document histograms to learn categories,
- Build visual vocabularies by quantizing image features: Compressed Histogram of Gradient descriptors [8].

$\rightarrow$ *histogram centroids*

$w_h = \sum_{i=1}^{d} h^i$: cumulative sum of bin values
$\tilde{}$: normalization operator

Why Jeffreys divergence?

Distance between two frequency histograms $\tilde{p}$ and $\tilde{q}$:
Kullback-Leibler divergence or *relative entropy*.

$$\begin{aligned}
\mathrm{KL}(\tilde{p}:\tilde{q}) &= H^{\times}(\tilde{p}:\tilde{q}) - H(\tilde{p}), \\
H^{\times}(\tilde{p}:\tilde{q}) &= \sum_{i=1}^{d} \tilde{p}^i \log \frac{1}{\tilde{q}^i}, \mathrm{cross-entropy} \\
H(\tilde{p}) &= H^{\times}(\tilde{p}:\tilde{p}) = \sum_{i=1}^{d} \tilde{p}^i \log \frac{1}{\tilde{p}^i}, \mathrm{Shannon\ entropy}.
\end{aligned}$$

$\rightarrow$ expected extra number of bits per datum that must be transmitted when using the "wrong" distribution $\tilde{q}$ instead of the true distribution $\tilde{p}$.
$\tilde{p}$ is hidden by nature (and hypothesized), $\tilde{q}$ is estimated.

Why Jeffreys divergence?

When clustering histograms, all histograms play the *same role* $\rightarrow$ Jeffreys [18] divergence:

$$\begin{aligned} J(p,q) &= \mathrm{KL}(p:q) + \mathrm{KL}(q:p), \\ J(p,q) &= \sum_{i=1}^{d} (p^i - q^i) \log \frac{p^i}{q^i} = J(q,p). \end{aligned}$$

$\rightarrow$ symmetrizes the KL divergence.
(aka. J-divergence or symmetrical Kullback-Leibler divergence, etc.)

$$\mathrm{KL}(p:q) = \sum_i p_i \log \frac{p_i}{q_i} + q_i - p_i$$

Jeffreys centroids: frequency and positive centroids

A set $\mathcal{H} = \{h_1, ..., h_n\}$ of *weighted histograms*.

$$c = \arg\min_x \sum_{j=1}^{n} \pi_j J(h_j, x),$$

$\pi_j > 0$'s histogram positive weights: $\sum_{j=1}^{n} \pi_j = 1$.

- Jeffreys positive centroid c:

$$c = \arg\min_{x \in \mathbb{R}_+^d} \sum_{j=1}^{n} \pi_j J(h_j, x),$$

- Jeffreys frequency centroid $\tilde{c}$:

$$\tilde{c} = \arg\min_{x \in \Delta_d} \sum_{j=1}^{n} \pi_j J(\tilde{h}_j, x),$$

Δ_d: Probability $(d-1)$-dimensional simplex.

Prior work

- Histogram clustering wrt. χ^2 distance [20]
- Histogram clustering wrt. Bhattacharyya distance [26, 33]
- Histogram clustering wrt. Kullback-Leibler distance as Bregman k-means clustering [1]
- Jeffreys frequency centroid [38] (Newton numerical optimization)
- Jeffreys frequency centroid as equivalent symmetrized Bregmen centroid [36]
- Mixed Bregman clustering [37]
- Smooth family of KL symmetrized centroids including Jensen-Shannon centroids and Jeffreys centroids in *limit* case [29]

Jeffreys positive centroid

$$c = \arg\min_{x \in \mathbb{R}_+^d} J(\mathcal{H}, x) = \arg\min_{x \in \mathbb{R}_+^d} \sum_{j=1}^{n} \pi_j J(h_j, x).$$

Theorem (Theorem 1)

The Jeffreys positive centroid $c = (c^1, ..., c^d)$ *of a set* $\{h_1, ..., h_n\}$ *of* n *weighted positive histograms with* d *bins can be calculated component-wise* exactly *using the Lambert W analytic function:*

$$c^i = \frac{a^i}{W(\frac{a^i}{g^i} e)},$$

where $a^i = \sum_{j=1}^{n} \pi_j h_j^i$ *denotes the coordinate-wise arithmetic weighted means and* $g^i = \prod_{j=1}^{n} (h_j^i)^{\pi_j}$ *the coordinate-wise geometric weighted means.*

Lambert analytic function [2] $W(x)e^{W(x)} = x$ for $x \geq 0$.

Jeffreys positive centroid (proof)

$$\min_x \quad \sum_{j=1}^{n} \pi_j J(h_j, x)$$

$$\min_x \quad \sum_{j=1}^{n} \pi_j \sum_{i=1}^{d} (h_j^i - x^i)(\log h_j^i - \log x^i)$$

$$\equiv \min_x \quad \sum_{i=1}^{d} \sum_{j=1}^{n} \pi_j (x^i \log x^i - x^i \log h_j^i - h_j^i \log x^i)$$

$$\sum_{i=1}^{d} x^i \log x^i - x^i \log \underbrace{\prod_{j=1}^{n} (h_j^i)^{\pi_j}}_{g} - \underbrace{\sum_{j=1}^{n} \pi_j h_j^i}_{a} \log x^i$$

$$\min_x \sum_{i=1}^{d} x^i \log \frac{x^i}{g} - a \log x^i$$

Jeffreys positive centroid (proof)

Coordinate-wise minimize:

$$\min_x x \log \frac{x}{g} - a \log x$$

Setting the derivative to zero, we solve:

$$\log \frac{x}{g} + 1 - \frac{a}{x} = 0$$

and get

$$x = \frac{a}{W(\frac{a}{g}e)}$$

Jeffreys frequency centroid: A guaranteed approximation

$$\tilde{c} = \arg\min_{x \in \Delta_d} \sum_{j=1}^{n} \pi_j J(\tilde{h}_j, x),$$

Relaxing x from probability simplex Δ_d to $\mathbb{R}_+^d$, we get

$$\tilde{c}' = \frac{c}{w_c}, c^i = \frac{a^i}{W(\frac{a^i}{g^i} e)}, w_c = \sum_i c^i$$

Lemma (Lemma 1)

The cumulative sum w_c of the bin values of the Jeffreys positive centroid c of a set of frequency histograms is less or equal to one:

$$0 < w_c \leq 1.$$

Proof of Lemma 1

From Theorem 1:

$$w_c = \sum_{i=1}^{d} c^i = \sum_{i=1}^{d} \frac{a^i}{W(\frac{a^i}{g^i} e)}.$$

Arithmetic-geometric mean inequality: $a^i \geq g^i$
Therefore $W(\frac{a^i}{g^i} e) \geq 1$ and $c^i \leq a^i$. Thus

$$w_c = \sum_{i=1}^{d} c^i \leq \sum_{i=1}^{d} a^i = 1$$

Lemma 2

Lemma (Lemma 2)

For any histogram x and frequency histogram $\tilde{h}$, we have $J(x, \tilde{h}) = J(\tilde{x}, \tilde{h}) + (w_x - 1)(\mathrm{KL}(\tilde{x} : \tilde{h}) + \log w_x)$, where w_x denotes the normalization factor ($w_x = \sum_{i=1}^{d} x^i$).

$$J(x, \tilde{H}) = J(\tilde{x}, \tilde{H}) + (w_x - 1)(\mathrm{KL}(\tilde{x} : \tilde{H}) + \log w_x),$$

where $J(x, \tilde{H}) = \sum_{j=1}^{n} \pi_j J(x, \tilde{h}_j)$ and
$\mathrm{KL}(\tilde{x} : \tilde{H}) = \sum_{j=1}^{n} \pi_j \mathrm{KL}(\tilde{x}, \tilde{h}_j)$ (with $\sum_{j=1}^{n} \pi_j = 1$).

Proof of Lemma 2

$$x^i = w_x \tilde{x}^i$$

$$J(x, \tilde{h}) = \sum_{i=1}^{d} (w_x \tilde{x}^i - \tilde{h}^i) \log \frac{w_x \tilde{x}^i}{\tilde{h}^i}$$

$$\begin{aligned}
J(x, \tilde{h}) &= \sum_{i=1}^{d} (w_x \tilde{x}^i \log \frac{\tilde{x}^i}{\tilde{h}^i} + w_x \tilde{x}^i \log w_x + \tilde{h}^i \log \frac{\tilde{h}^i}{\tilde{x}^i} - \tilde{h}^i \log w_x) \\
&= (w_x - 1) \log w_x + J(\tilde{x}, \tilde{h}) + (w_x - 1) \sum_{i=1}^{d} \tilde{x}^i \log \frac{\tilde{x}^i}{\tilde{h}^i} \\
&= J(\tilde{x}, \tilde{h}) + (w_x - 1)(\mathrm{KL}(\tilde{x} : \tilde{h}) + \log w_x)
\end{aligned}$$

since $\sum_{i=1}^{d} \tilde{h}^i = \sum_{i=1}^{d} \tilde{x}^i = 1$.

Guaranteed approximation of $\tilde{c}$

Theorem (Theorem 2)

Let $\tilde{c}$ denote the Jeffreys frequency centroid and $\tilde{c}' = \frac{c}{w_c}$ the normalized Jeffreys positive centroid. Then the approximation factor $\alpha_{\tilde{c}'} = \frac{J(\tilde{c}', \tilde{H})}{J(\tilde{c}, \tilde{H})}$ is such that $1 \leq \alpha_{\tilde{c}'} \leq \frac{1}{w_c}$ (with $w_c \leq 1$).

Proof of Theorem 2

$$J(c, \tilde{H}) \leq J(\tilde{c}, \tilde{H}) \leq J(\tilde{c}', \tilde{H})$$

From Lemma 2, since
$J(\tilde{c}', \tilde{H}) = J(c, \tilde{H}) + (1 - w_c)(\mathrm{KL}(\tilde{c}', \tilde{H}) + \log w_c))$ and
$J(c, \tilde{H}) \leq J(\tilde{c}, \tilde{H})$

$$1 \leq \alpha_{\tilde{c}'} \leq 1 + \frac{(1 - w_c)(\mathrm{KL}(\tilde{c}', \tilde{H}) + \log w_c)}{J(\tilde{c}, \tilde{H})}$$

$$\mathrm{KL}(\tilde{c}' : \tilde{H}) = \frac{1}{w_c}\mathrm{KL}(c, \tilde{H}) - \log w_c$$

$$\alpha_{\tilde{c}'} \leq 1 + \frac{(1 - w_c)\mathrm{KL}(c, \tilde{H})}{w_c J(\tilde{c}, \tilde{H})}$$

Since $J(\tilde{c}, \tilde{H}) \geq J(c, \tilde{H})$ and $\mathrm{KL}(c, \tilde{H}) \leq J(c, \tilde{H})$, we get
$\alpha_{\tilde{c}'} \leq \frac{1}{w_c}$.
When $w_c = 1$ the bound is tight.

In practice...

c in closed-form $\rightarrow$ compute w_c, $\mathrm{KL}(c, \tilde{H})$, $J(c, \tilde{H})$.
Bound the approximation factor $\alpha_{\tilde{c}'}$ as:

$$\alpha_{\tilde{c}'} \leq 1 + \left(\frac{1}{w_c} - 1 \right) \frac{\mathrm{KL}(c, \tilde{H})}{J(c, \tilde{H})} \leq \frac{1}{w_c}$$

Fine approximation

From [38, 36], minimization of Jeffreys frequency centroid equivalent to:

$$\tilde{c} = \arg\min_{\tilde{x} \in \Delta_d} \mathrm{KL}(\tilde{a} : \tilde{x}) + \mathrm{KL}(\tilde{x} : \tilde{g})$$

Lagrangian function enforcing $\sum_i c^i = 1$:

$$\log \frac{\tilde{c}^i}{\tilde{g}^i} + 1 - \frac{\tilde{a}^i}{\tilde{c}^i} + \lambda = 0$$

$$\tilde{c}^i = \frac{\tilde{a}^i}{W\left(\frac{\tilde{a}^i e^{\lambda+1}}{\tilde{g}^i}\right)}$$

$\lambda = -\mathrm{KL}(\tilde{c} : \tilde{g}) \leq 0$

Fine approximation: Bisection search

$$c^i \leq 1 \Rightarrow c^i = \frac{\tilde{a}^i}{W\left(\frac{\tilde{a}^i e^{\lambda+1}}{\tilde{g}^i}\right)} \leq 1$$

$$\lambda \geq \log(e^{\tilde{a}^i}\tilde{g}^i) - 1 \forall i, \quad \lambda \in [\max_i \log(e^{\tilde{a}^i}\tilde{g}^i) - 1, 0]$$

$$s(\lambda) = \sum_i c^i(\lambda) = \sum_{i=1}^{d} \frac{\tilde{a}^i}{W\left(\frac{\tilde{a}^i e^{\lambda+1}}{\tilde{g}^i}\right)}$$

Function s: monotonously decreasing with $s(0) \leq 1$.
$\rightarrow$ Bisection search for $s(\lambda^*) \simeq 1$ for arbitrary precision.

Experiments: Caltech-256

Caltech-256 [16]: 30607 images labeled into 256 categories (256 Jeffreys centroids).
Arbitrary floating-point precision: `http://www.apfloat.org/`

$$\tilde{c}'' = \frac{\tilde{a} + \tilde{g}}{2}$$

	α_c(optimal positive)	$\alpha_{\tilde{c}'}$(n′lized approx.)	$w_c \leq 1$(n′lizing coeff.t)	$\alpha_{\tilde{c}''}$(Veldhuis' approx.)
avg	0.9648680345638155	**1.0002205080964255**	0.9338228644308926	1.065590178484613
min	0.906414219584823	**1.0000005079528809**	0.8342819488534723	1.0027707382095195
max	0.9956399220678585	**1.0000031489541772**	0.9931975105809021	1.3582296675397754

Experiments: Synthetic data-sets

Random binary histograms

$$\alpha = \frac{J(\tilde{c}')}{J(\tilde{c})} \geq 1$$

Performance:
$\bar{\alpha} \sim 1.0000009, \alpha_{\max} \sim 1.00181506, \alpha_{\min} = 1.000000.$

Express better worst-case upper bound performance?

Summary and conclusion

- Jeffreys positive centroid c in closed-form
- normalized Jeffreys positive centroid $\tilde{c}'$ within approximation factor $\frac{1}{w_c}$
- Bisection search for arbitrary fine approximation of $\tilde{c}$.

$\rightarrow$ Variational Jeffreys k-means clustering

Other Kullback-Leibler symmetrizations:

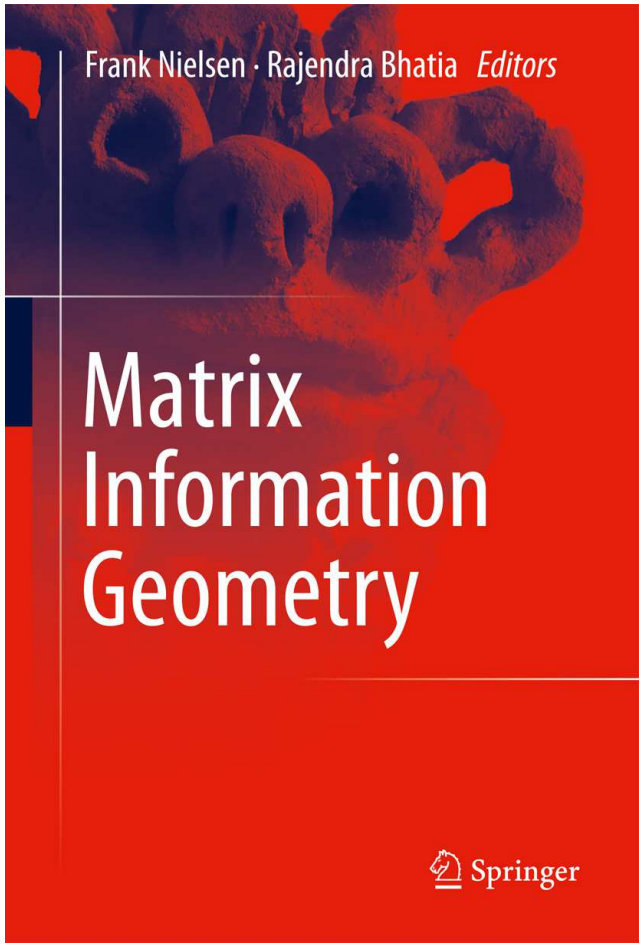
- Jensen-Shannon divergence [19]
- Chernoff divergence [9]
- Family of symmetrized centroids including Jensen-Shannon and Jeffreys centroids [29]
- Infinitely many symmetrizations using quasi-arithmetic means

Clustering: A never-ending story...

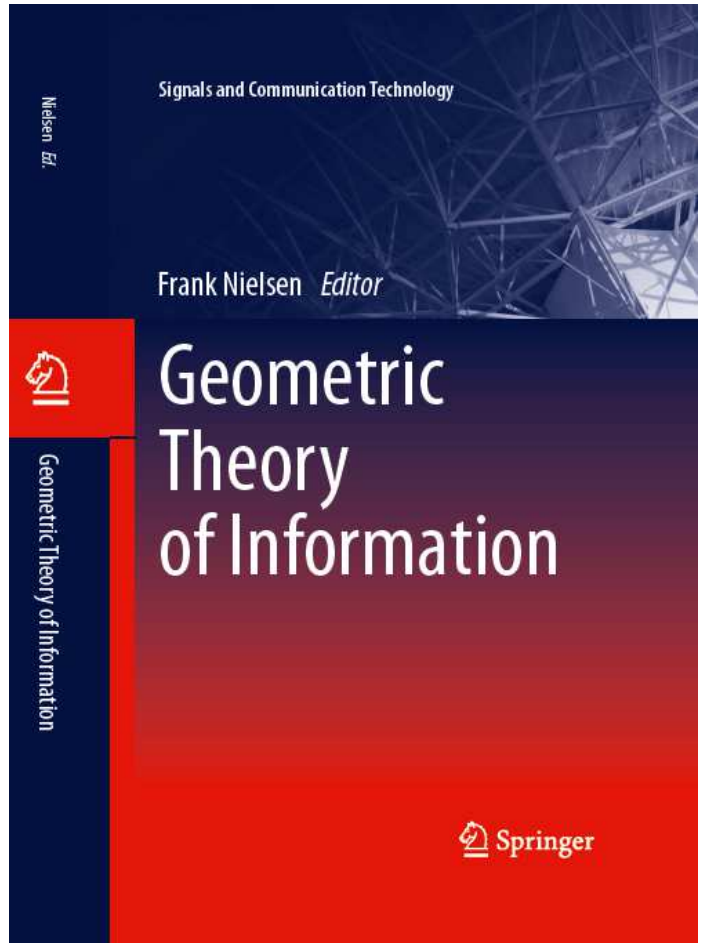
An old problem with many new recent results!

21st century is the revolution of data science

Computational Information Geometry



[32]

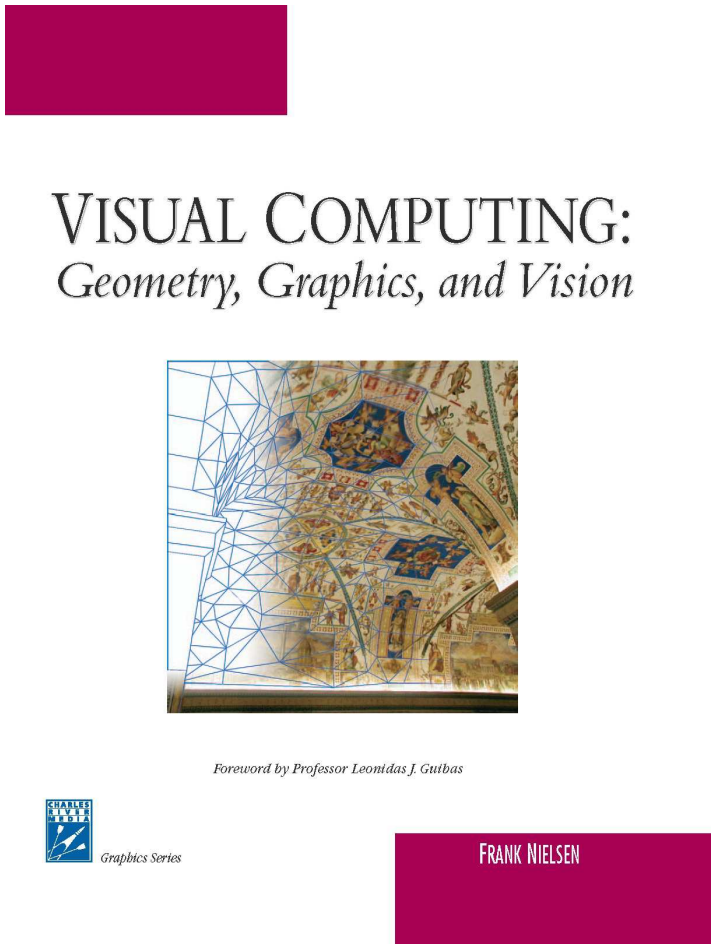


[31]

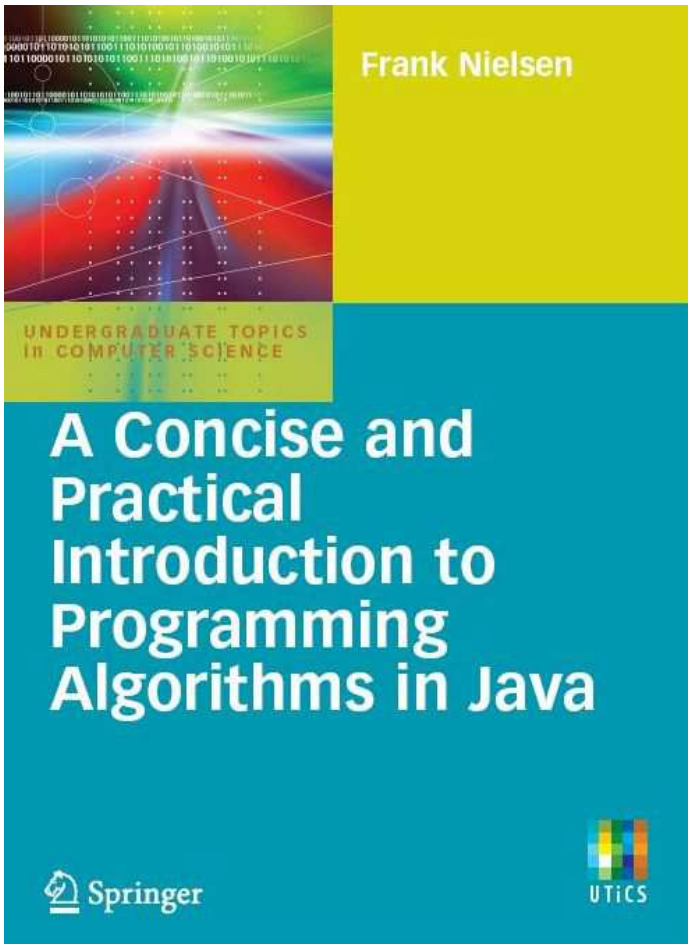
http://www.springer.com/engineering/signals/book/978-3-642-30231-2
http://www.sonycsl.co.jp/person/nielsen/infogeo/MIG/MIGBOOKWEB/

http://www.springer.com/engineering/signals/book/978-3-319-05316-5
http://www.sonycsl.co.jp/person/nielsen/infogeo/GTI/GeometricTheoryOfInformation.html

Textbooks: Visual computing



[27]



[28]

http://www.sonycsl.co.jp/person/nielsen/visualcomputing/

http://www.lix.polytechnique.fr/~nielsen/JavaProgramming/

Bibliography I


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


D. A. Barry, P. J. Culligan-Hensley, and S. J. Barry.
Real values of the W-function.
ACM Trans. Math. Softw., 21(2):161–171, June 1995.


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


Anup Bhattacharya, Ragesh Jaiswal, and Nir Ailon.
A tight lower bound instance for k-means++ in constant dimension.
CoRR, abs/1401.2912, 2014.


Brigitte Bigi.
Using Kullback-Leibler distance for text categorization.
In *Proceedings of the 25th European conference on IR research (ECIR)*, ECIR'03, pages 305–319, Berlin, Heidelberg, 2003. Springer-Verlag.


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.

Bibliography II

Vijay Chandrasekhar, Gabriel Takacs, David M. Chen, Sam S. Tsai, Yuriy A. Reznik, Radek Grzeszczuk, and Bernd Girod.
Compressed histogram of gradients: A low-bitrate descriptor.
International Journal of Computer Vision, 96(3):384–399, 2012.

Herman Chernoff.
A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations.
Annals of Mathematical Statistics, 23:493–507, 1952.

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.

G. Csurka, C. Bray, C. Dance, and L. Fan.
Visual categorization with bags of keypoints.
Workshop on Statistical Learning in Computer Vision (ECCV), pages 1–22, 2004.

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

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

Edward W. Forgy.
Cluster analysis of multivariate data: efficiency vs interpretability of classifications.
Biometrics, 1965.

Bibliography III

Vincent Garcia and Frank Nielsen.

Simplification and hierarchical representations of mixtures of exponential families.
Signal Processing, 90(12):3197–3212, 2010.

G. Griffin, A. Holub, and P. Perona.

Caltech-256 object category dataset.
Technical Report 7694, California Institute of Technology, 2007.

John A. Hartigan.

Clustering Algorithms.
John Wiley & Sons, Inc., New York, NY, USA, 99th edition, 1975.

Harold Jeffreys.

An invariant form for the prior probability in estimation problems.
Proceedings of the Royal Society of London, 186(1007):453–461, March 1946.

Jianhua Lin.

Divergence measures based on the Shannon entropy.
IEEE Transactions on Information Theory, 37:145–151, 1991.

Huan Liu and Rudy Setiono.

Chi2: Feature selection and discretization of numeric attributes.
In *Proceedings of the Seventh International Conference on Tools with Artificial Intelligence (TAI)*, pages 88–, Washington, DC, USA, 1995. IEEE Computer Society.

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.

Bibliography IV

Stuart P. Lloyd.

Least squares quantization in PCM.
Technical report, Bell Laboratories, 1957.

James B. MacQueen.

Some methods of classification and analysis of multivariate observations.
In L. M. Le Cam and J. Neyman, editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, Berkeley, CA, USA, 1967.

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.

Max Mignotte.

Segmentation by fusion of histogram-based k-means clusters in different color spaces.
IEEE Transactions on Image Processing (TIP), 17(5):780–787, 2008.

Frank Nielsen.

Visual Computing: Geometry, Graphics, and Vision.
Charles River Media / Thomson Delmar Learning, 2005.

Frank Nielsen.

A Concise and Practical Introduction to Programming Algorithms in Java.
Undergraduate Topics in Computer Science (UTiCS). Springer Verlag, 2009.
http://www.springer.com/computer/programming/book/978-1-84882-338-9.

Bibliography V


Frank Nielsen.
A family of statistical symmetric divergences based on Jensen's inequality.
CoRR, abs/1009.4004, 2010.


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


Frank Nielsen.
Geometric Theory of Information.
Springer, 2014.


Frank Nielsen and Rajendra Bhatia, editors.
Matrix Information Geometry (Revised Invited Papers). Springer, 2012.


Frank Nielsen and Sylvain Boltz.
The Burbea-Rao and Bhattacharyya centroids.
IEEE Transactions on Information Theory, 57(8):5455–5466, August 2011.


Frank Nielsen and Richard Nock.
Clustering multivariate normal distributions.
In *Emerging Trends in Visual Computing*, pages 164–174. Springer, 2009.


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


Frank Nielsen and Richard Nock.
Sided and symmetrized Bregman centroids.
IEEE Transactions on Information Theory, 55(6):2048–2059, June 2009.

Bibliography VI

Richard Nock, Panu Luosto, and Jyrki Kivinen.

Mixed Bregman clustering with approximation guarantees.
In *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases*, pages 154–169, Berlin, Heidelberg, 2008. Springer-Verlag.

Raymond N. J. Veldhuis.

The centroid of the symmetrical Kullback-Leibler distance.
IEEE signal processing letters, 9(3):96–99, March 2002.

Haizhou Wang and Mingzhou Song.

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

Juanying Xie, Shuai Jiang, Weixin Xie, and Xinbo Gao.

An efficient global k-means clustering algorithm.
Journal of computers, 6(2), 2011.