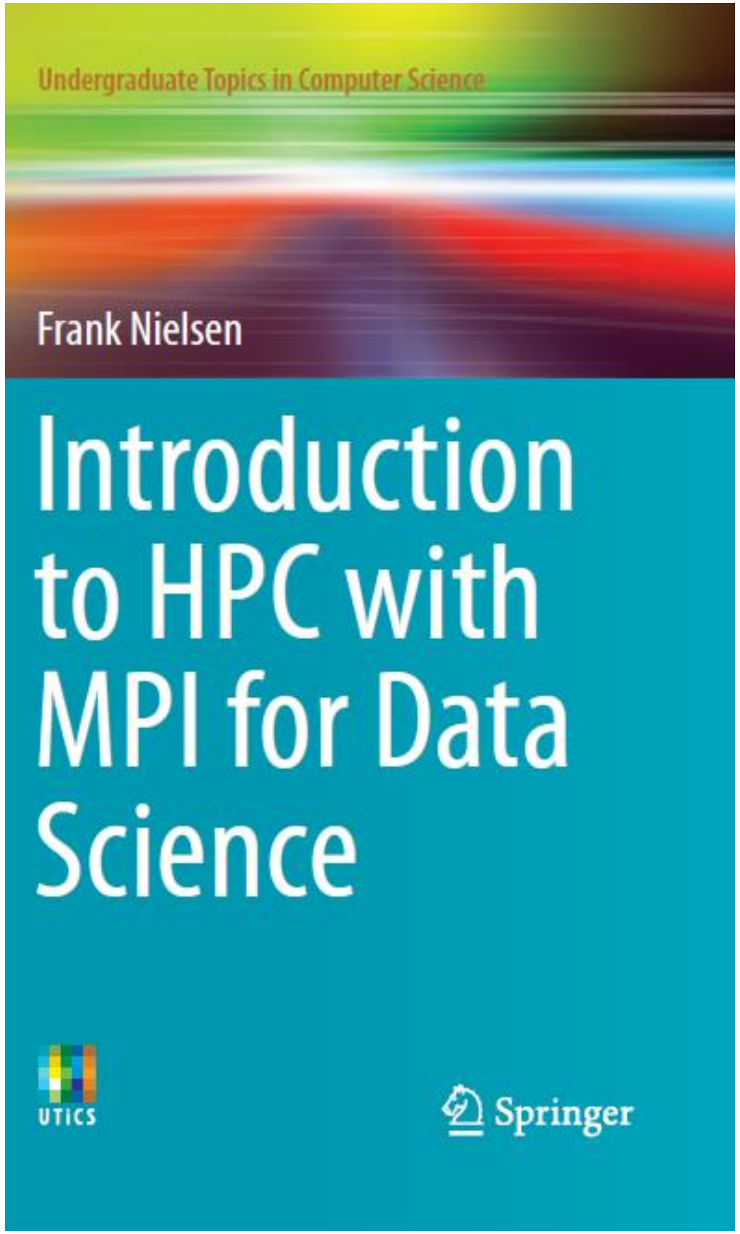
Research cards

Frank Nielsen

FrankNielsen.github.com

@FrnkNlsn



- Preface
- **Part I. High Performance Computing (HPC) with the Message Passing Interface (MPI)**
  - A glance at High Performance Computing (HPC)
  - Introduction to MPI: The Message Passing Interface
  - Topology of interconnection networks
  - Parallel Sorting
  - Parallel linear algebra
  - The MapReduce paradigm
- **Part II. High Performance Computing (HPC) for Data Science (DS)**
  - Partition-based clustering with k-means
  - Hierarchical clustering
  - Supervised learning: Practice and theory of classification with the k-NN rule
  - Fast approximate optimization in high dimensions with core-sets and fast dimension reduction
  - Parallel algorithms for graphs
- Appendices
  - Written exam
  - SLURM: A resource manager & job scheduler on clusters of machines

@FrnkNlsn

https://franknielsen.github.io/HPC4DS/index.html

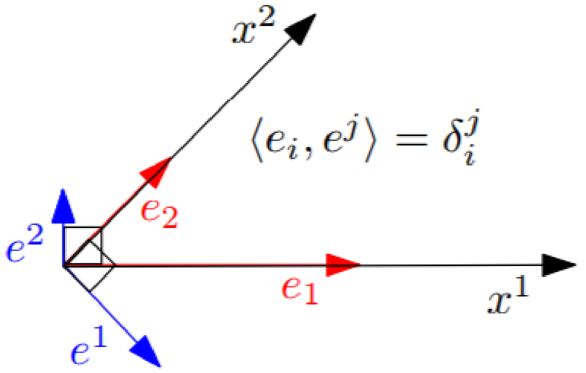
Metric tensor g: Raising/lowering vector indices

- Vectors v are **geometric objects**, independent of any coordinate systems.
- A vector is written in *any* basis $B_1, \ldots, B_n$ using corresponding **components**:

$$[v]_{B_1}, [v]_{B_2}, \ldots, [v]_{B_n}$$

We write the components using column "vectors" for algebra operations

- Vector components in primal basis B are $[v]_B = \begin{bmatrix} v^1 \\ \vdots \\ v^d \end{bmatrix}$ (contravariant, upper index) and in **reciprocal basis** B^* are $[v]_{B^*} = \begin{bmatrix} v_1 \\ \vdots \\ v_d \end{bmatrix}$ (covariant, lower index).



- **Metric tensor** g is a **bilinear form**, positive-definite (2-covariant tensor)

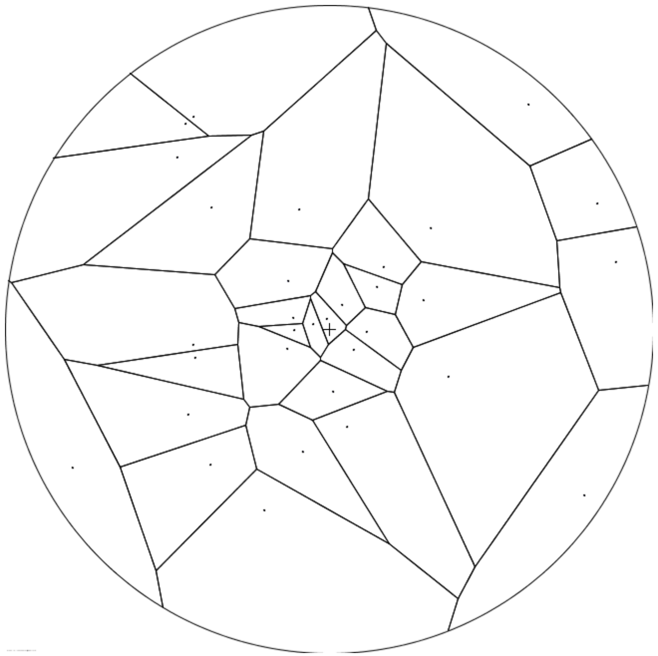
$$g(v,w) = \langle v,w \rangle_g = [v]_B^T [g]_B \, [w]_B = [v]_B^T [w]_{B^*} = [v]_{B^*}^T [w]_B$$

- Algebra: $[v]_{B^*} = [g]_B [v]_B$ (lowering index) and $[v]_B = [g]_{B^*} [v]_{B^*}$ (raising index)
- Algebraic identity: $[g]_{B^*} [g]_B$=I, the identity matrix

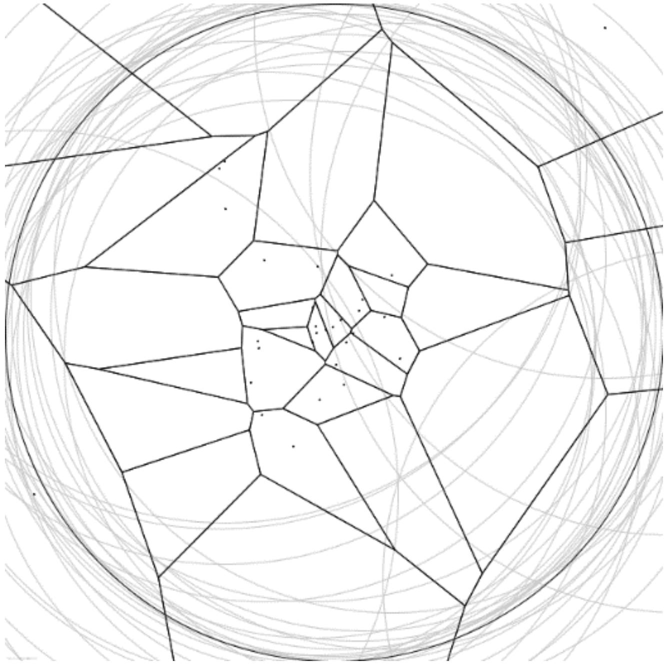
An elementary introduction to information geometry, https://arxiv.org/abs/1808.08271

Hyperbolic Voronoi diagram (HVD)

- In Klein ball model, bisectors are hyperplanes clipped by the unit ball
- Klein Voronoi diagram is equivalent to a **clipped power diagram**



Klein hyperbolic Voronoi diagram
(all cells non-empty)



Power diagram (additive weights)
(some cells may be empty)

Hyperbolic Voronoi diagrams made easy, https://arxiv.org/abs/0903.3287

Visualizing Hyperbolic Voronoi Diagrams, https://www.youtube.com/watch?v=i9IUzNxeH4o

Fast approximation of the Löwner extremal matrix

Problem 1 (Löwner maximal matrices)

$$\bar{S} = \inf\{X \in \mathrm{Sym}(\mathbb{R}) \ : \ \forall i \in [n], X \succeq S_i\}$$

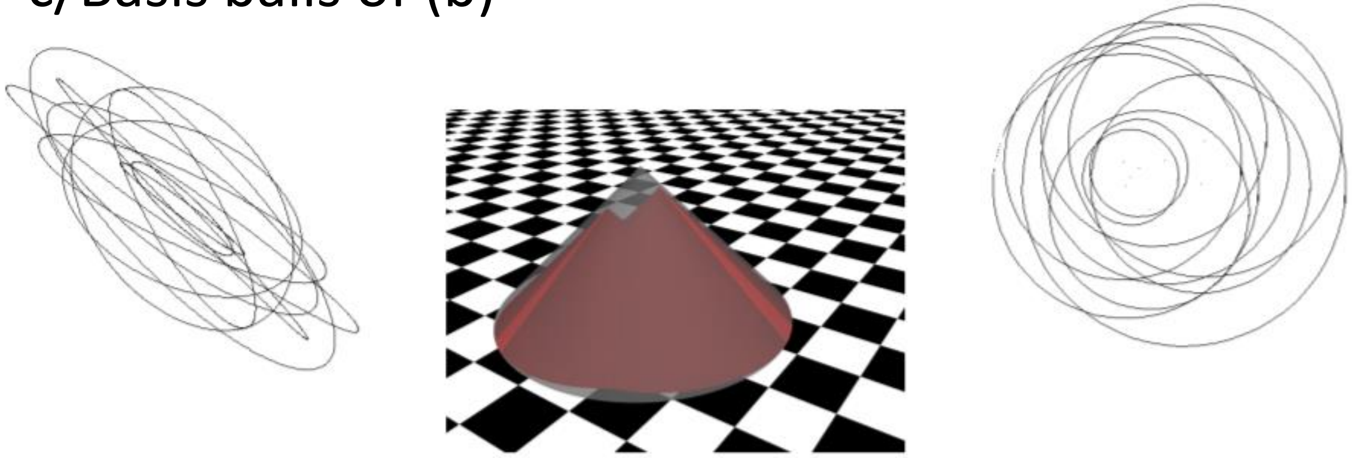
Finding the extremal matrix of positive-definite matrices amount to compute the smallest enclosing ball of cone basis balls

<u>Visualizations of a positive-definite matrix:</u>

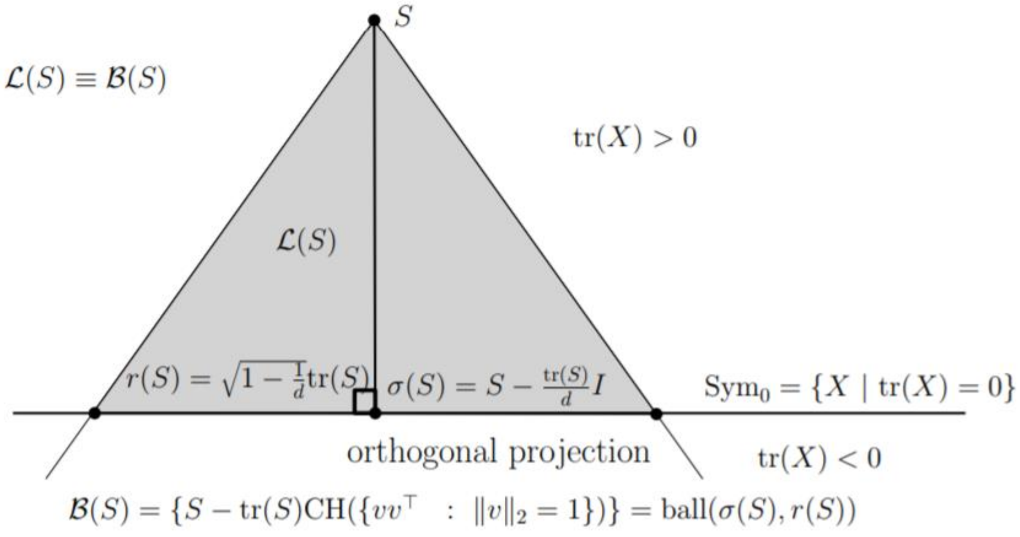
a/Covariance ellipsoids

b/Translated positive-define cone

c/Basis balls of (b)



(a) (b) (c)



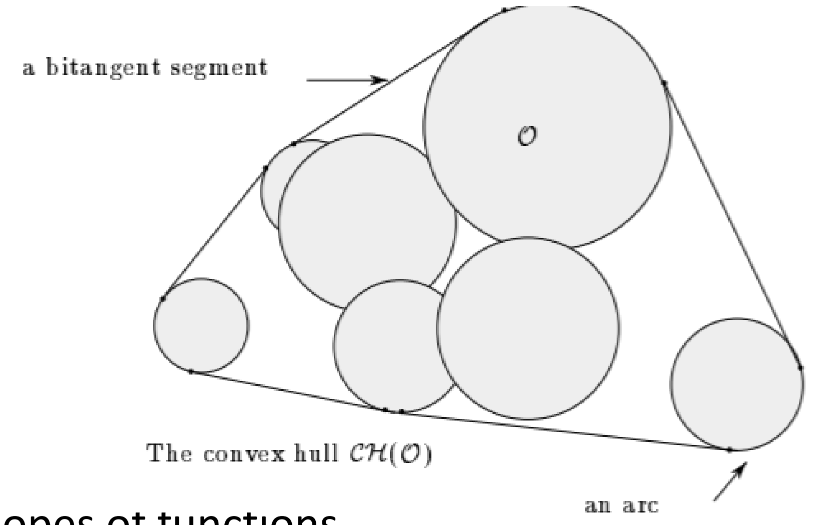
https://arxiv.org/abs/1604.01592

Output-sensitive convex hull construction of 2D objects

N objects, boundaries intersect pairwise in at most m points

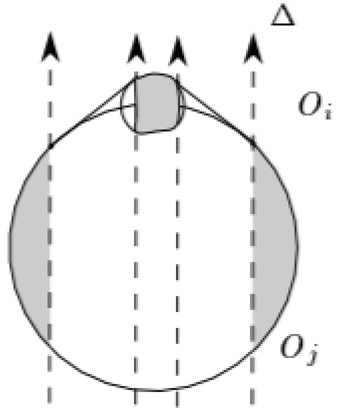
Convex hull of disks (m=2), of ellipses (m=4), etc.

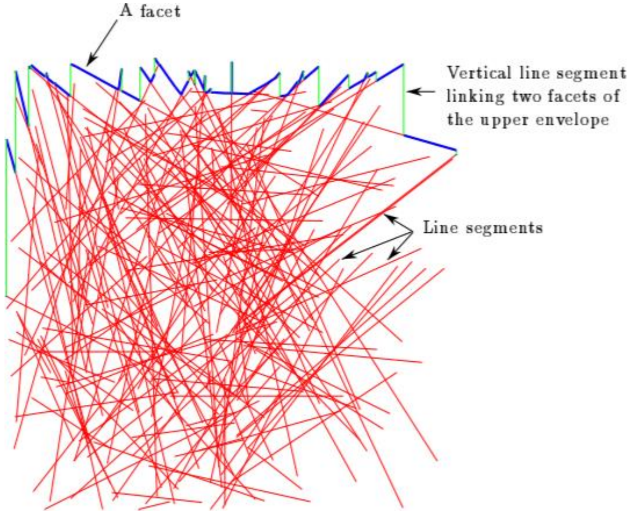
Complexity bounded using Ackermann's inverse function α



$m \backslash \lambda(n,m)$	Lower Bound Ω	Upper Bound O
1	n	n
2	$2n-1$	$2n-1$
3	$\Omega(n \times \alpha(n))$	$O(n \times \alpha(n))$
4	$\Omega(n \times 2^{\alpha(n)})$	$O(n \times 2^{\alpha(n)})$
$2s+1$	$\Omega(n \times 2^{O(\alpha(n)^{s-1})})$	$O(n \times \alpha(n)^{O(\alpha(n)^{s-1})})$
$2s+2$	$\Omega(n \times 2^{O(\alpha(n)^{s})})$	$O(n \times 2^{O(\alpha(n)^{s})})$

Extend to upper envelopes of functions pairwise intersecting in m points





Output-Sensitive Convex Hull Algorithms of Planar Convex Objects, IJCGA (1998)

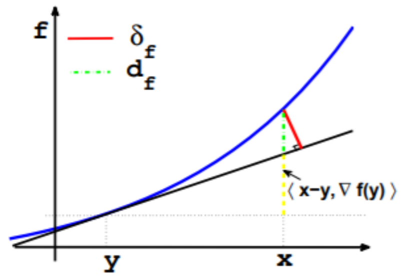
Shape Retrieval Using Hierarchical Total Bregman Soft Clustering

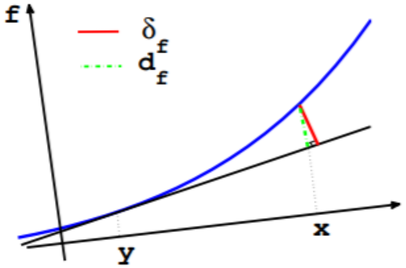
Definition *The total Bregman divergence δ associated with a real valued strictly convex and differentiable function f defined on a convex set X between points $x, y \in X$ is defined as,*

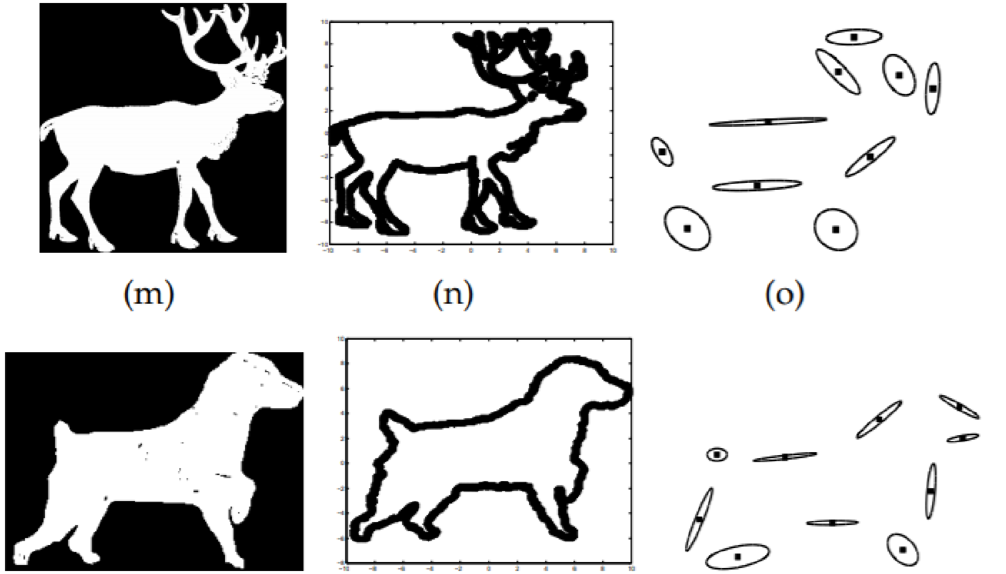
$$\delta_f(x,y) = \frac{f(x) - f(y) - \langle x - y, \nabla f(y)\rangle}{\sqrt{1 + \|\nabla f(y)\|^2}},$$

$\langle \cdot, \cdot \rangle$ is inner product and $\|\nabla f(y)\|^2 = \langle \nabla f(y), \nabla f(y)\rangle$ generally.

X	$f(x)$	$\delta_f(x,y)$	t-center	ℓ_1-norm BD center	Remark
$\mathbb{R}$	x^2	$\frac{(x-y)^2}{\sqrt{1+4y^2}}$	$\sum_i w_i x_i$	$\sum_i x_i$	total square loss (tSL)
$\mathbb{R} - \mathbb{R}_-$	$x \log x$	$\frac{x \log \frac{x}{y} + \bar{x} \log \frac{\bar{x}}{\bar{y}}}{\sqrt{1+y(1+\log y)^2+\bar{y}(1+\log \bar{y})^2}}$	$\prod_i (x_i)^{w_i}$	$\sum_i x_i$	
$[0,1]$	$-\log x$	$\frac{\frac{x}{y} - \log \frac{x}{y} - 1}{\sqrt{1+y^{-2}}}$	$\frac{\sum_i (x_i/(1-x_i))^{w_i}}{1+\sum_i (x_i/(1-x_i))^{w_i}}$	$\sum_i x_i$	total logistic loss
$\mathbb{R}_+$	$-\log x$	$\frac{\frac{x}{y} - \log \frac{x}{y} - 1}{\sqrt{1+y^{-2}}}$	$\frac{1}{\sum_i w_i/x_i}$	$\sum_i x_i$	total Itakura-Saito distance
$\mathbb{R}$	e^x	$\frac{e^x - e^y - (x-y)e^y}{\sqrt{1+e^{2y}}}$	$\sum_i w_i x_i$	$\sum_i x_i$	
$\mathbb{R}^d$	$\|x\|^2$	$\frac{\|x-y\|^2}{\sqrt{1+4\|y\|^2}}$	$\sum_i w_i x_i$	$\sum_i x_i$	total squared Euclidean
$\mathbb{R}^d$	$x^t A x$	$\frac{(x-y)^t A (x-y)}{\sqrt{1+4\|Ay\|^2}}$	$\sum_i w_i x_i$	$\sum_i x_i$	total Mahalanobis distance
Δ^d	$\sum_{j=1}^d x_j \log x_j$	$\frac{\sum_{j=1}^d x_j \log \frac{x_j}{y_j}}{\sqrt{1+\sum_{j=1}^d y_j (1+\log y_j)^2}}$	$c \prod_i (x_i)^{w_i}$	$\sum_i x_i$	total KL divergence (tKL)
$\mathbb{C}^{m \times n}$	$\|x\|_F^2$	$\frac{\|x-y\|_F^2}{\sqrt{1+4\|y\|_F^2}}$	$\frac{\|x-y\|_F^2}{\sqrt{1+4\|y\|_F^2}}$	$\sum_i x_i$	total squared Frobenius







(m) (n) (o)

t-center: $$\bar{x} = \arg\min_x \delta_f^1(x, E) = \arg\min_x \sum_{i=1}^n \delta_f(x, x_i)$$

Robust to noise/outliers

@FrnkNlsn

IEEE TPAMI 34, 2012

Total Bregman divergence and its applications to DTI analysis

IEEE Transactions on medical imaging, *30*(2), 475-483, 2010.

Definition *The total Bregman divergence (TBD) δ_f associated with a real valued strictly convex and differentiable function f defined on a convex set X between points $x, y \in X$ is defined as,*

$$\delta_f(x,y) = \frac{f(x) - f(y) - \langle x - y, \nabla f(y)\rangle}{\sqrt{1 + \|\nabla f(y)\|^2}}, \qquad (2)$$

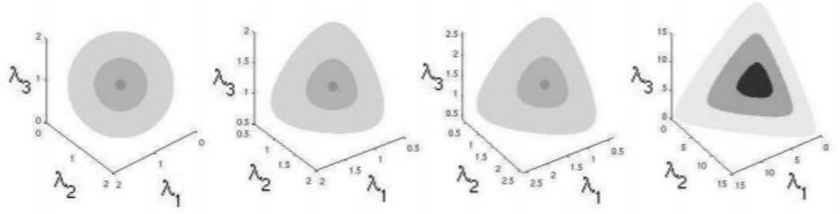
$\langle \cdot, \cdot \rangle$ is inner product as in definition II.1, and $\|\nabla f(y)\|^2 = \langle \nabla f(y), \nabla f(y)\rangle$ generally.

$$tKL(P,Q) = \frac{\int p \log \frac{p}{q} dx}{\sqrt{1 + \int (1 + \log q)^2 q dx}}$$

$$= \frac{\log(\det(P^{-1}Q)) + tr(Q^{-1}P) - n}{2\sqrt{c + \frac{(\log(\det Q))^2}{4} - \frac{n(1+\log 2\pi)}{2}\log(\det Q)}}$$

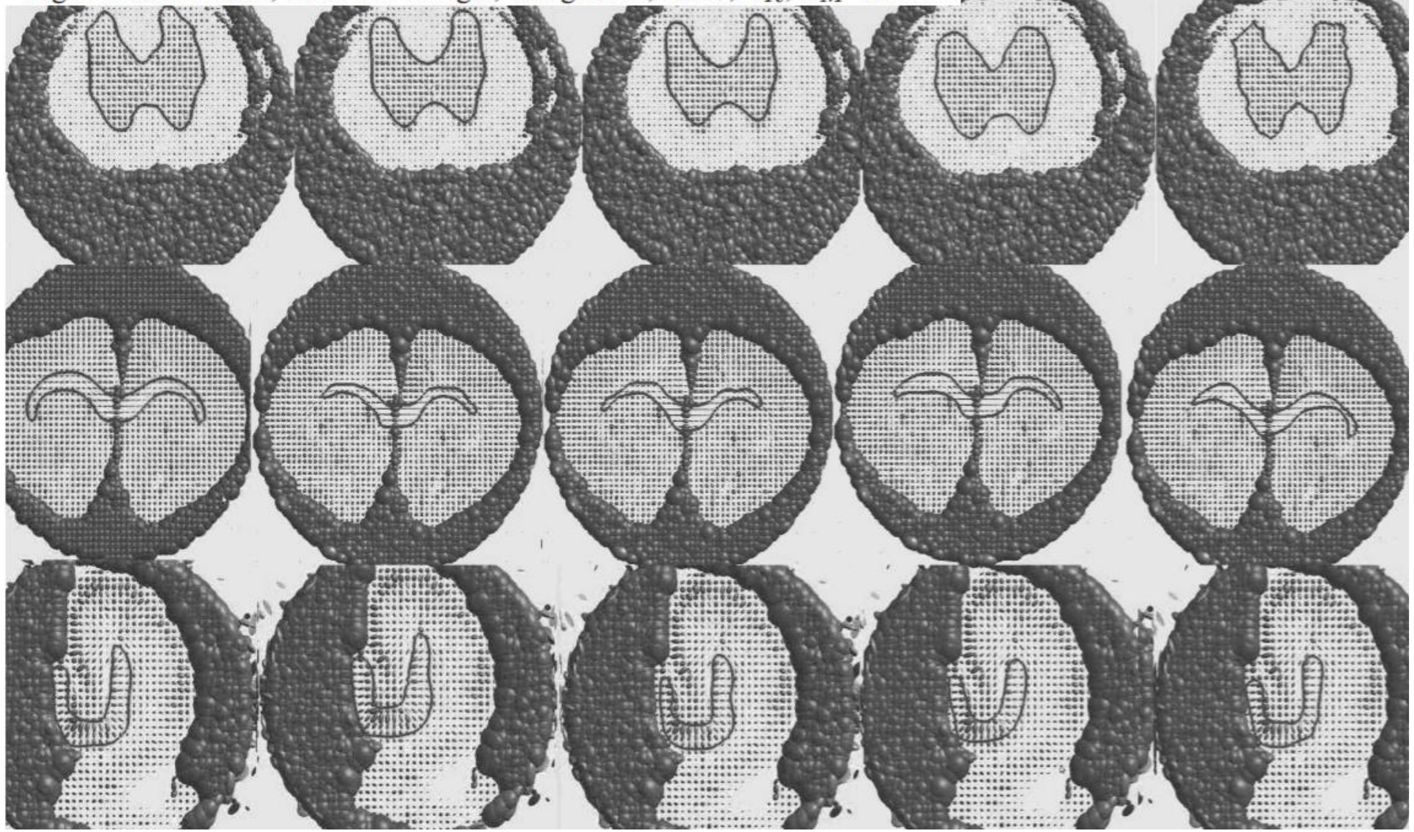
$$tKL(P,Q) = tKL(A'PA, A'QA), \quad \forall A \in SL(n),$$

$$tSL(P,Q) = \frac{\int (p-q)^2 dx}{\sqrt{1 + \int (2q)^2 q dx}} = \frac{1/\sqrt{\det(2P)} + 1/\sqrt{\det(2Q)} - 2/\sqrt{\det(P+Q)}}{(2\pi)^n + 4\sqrt{(2\pi)^n}/\sqrt{\det(3Q)}}$$



The isosurfaces of $d_F(P,I) = r$, $d_R(P,I) = r$, $KL_s(P,I) = r$ and $tKL(P,I) = r$ shown from left to right. The three axes are eigenvalues of P.

segmentation results, from left to right, using tKL, KL_s, d_R, d_M and LE.



@FrnkNlsn

k-MLE: Inferring statistical mixtures a la k-Means

arxiv:1203.5181

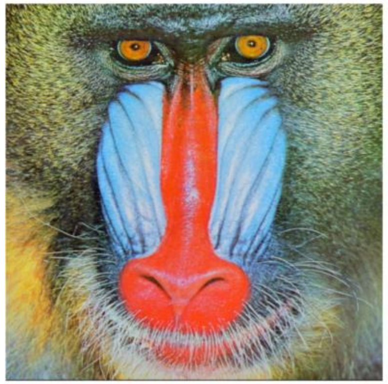
Bijection between regular Bregman divergences and regular (dual) exponential families

$$\log p_F(x;\theta) = -B_{F^*}(t(x):\eta) + F^*(t(x)) + k(x)$$

Maximum log-likelihood estimate (exp. Family) = dual Bregman centroid

$$\max_{\theta\in\mathbb{N}} \quad \bar{l}(\theta;x_1,...,x_n) = \frac{1}{n}\sum^n(\langle t(x_i),\theta\rangle - F(\theta) + k(x_i))$$

$$\equiv \min_{\eta\in\mathbb{M}} \quad \frac{1}{n}\sum_{i=1}^n B_{F^*}(t(x_i):\eta)$$



Exponential Family $p_F(x\|\theta)$	$\Leftrightarrow$	Dual Bregman divergence B_{F^*}
Spherical Gaussian	$\Leftrightarrow$	Squared Euclidean divergence
Multinomial	$\Leftrightarrow$	Kullback-Leibler divergence
Poisson	$\Leftrightarrow$	I-divergence
Geometric	$\Leftrightarrow$	Itakura-Saito divergence
Wishart	$\Leftrightarrow$	log-det/Burg matrix divergence

Classification Expectation-Maximization (CEM) yields a dual Bregman k-means for mixtures of exponential families (however, k-MLE is not consistent)

Online k-MLE for Mixture Modeling with Exponential Families, GSI 2015
On learning statistical mixtures maximizing the complete likelihood, AIP 2014
Hartigan's Method for k-MLE: Mixture Modeling with Wishart Distributions and Its Application to Motion Retrieval, GTI 2014
A New Implementation of k-MLE for Mixture Modeling of Wishart Distributions, GSI 2013
Fast Learning of Gamma Mixture Models with k-MLE, SIMBAD 2013
k-MLE: A fast algorithm for learning statistical mixture models, ICASSP 2012
k-MLE for mixtures of generalized Gaussians, ICPR 2012

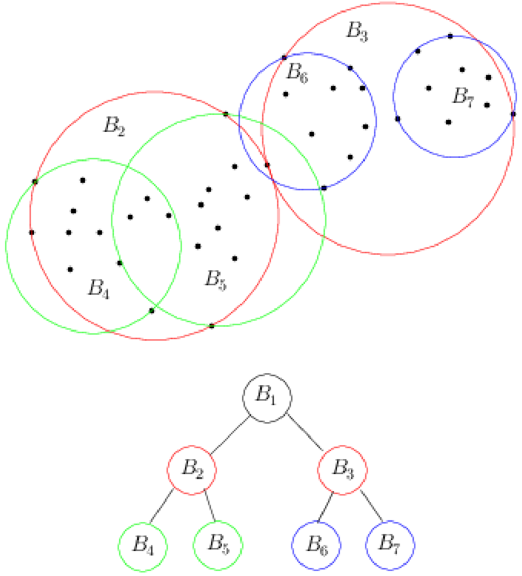
@FrnkNlsn

Fast Proximity queries for Bregman divergences (incl. KL)

Fast <u>Nearest Neighbour Queries</u> for Bregman divergences

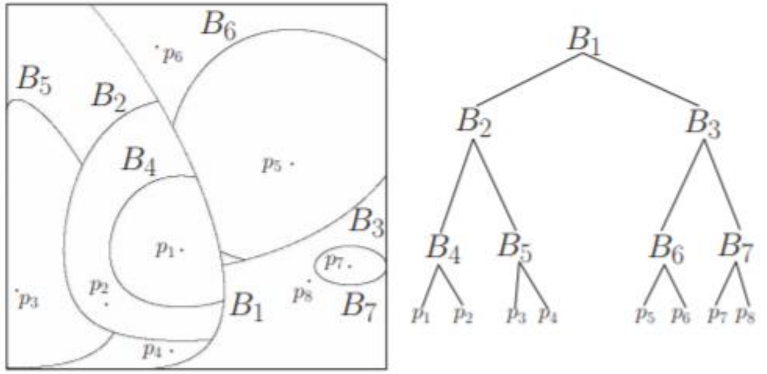
Space partition induced by

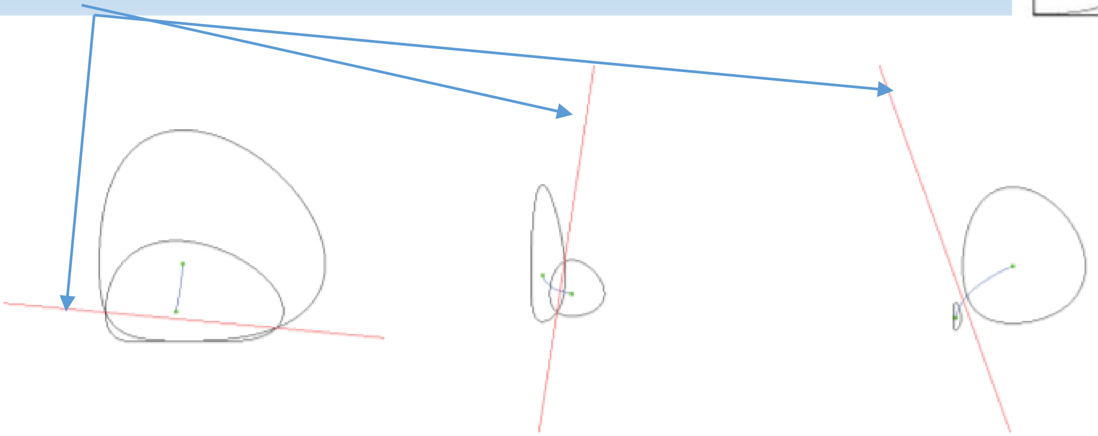
Bregman vantage point trees

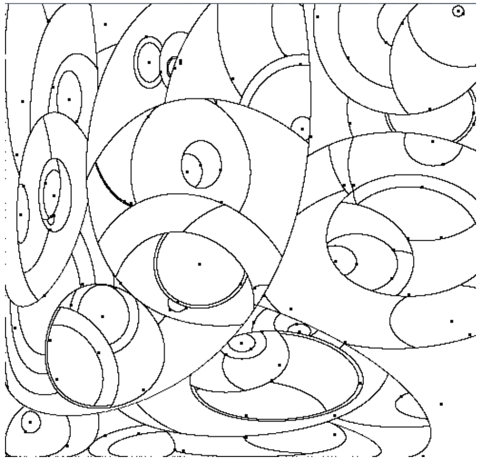


Bregman ball trees

Key property:
Check whether two Bregman spheres Intersect or not easily
(radical hyperplane, space of spheres)







E.g., Extended Kullback-Leibler

C++ source code https://www.lix.polytechnique.fr/~nielsen/BregmanProximity/

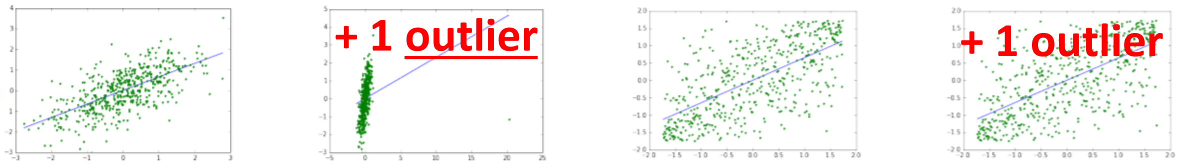
@FrnkNlsn

Bregman vantage point trees for efficient nearest Neighbor Queries, ICME 2009
Tailored Bregman ball trees for effective nearest neighbors, EuroCG 2009

Optimal Copula Transport: Clustering Time Series

Distance between random variables (Mutual Information, similarity: correlation coefficient)

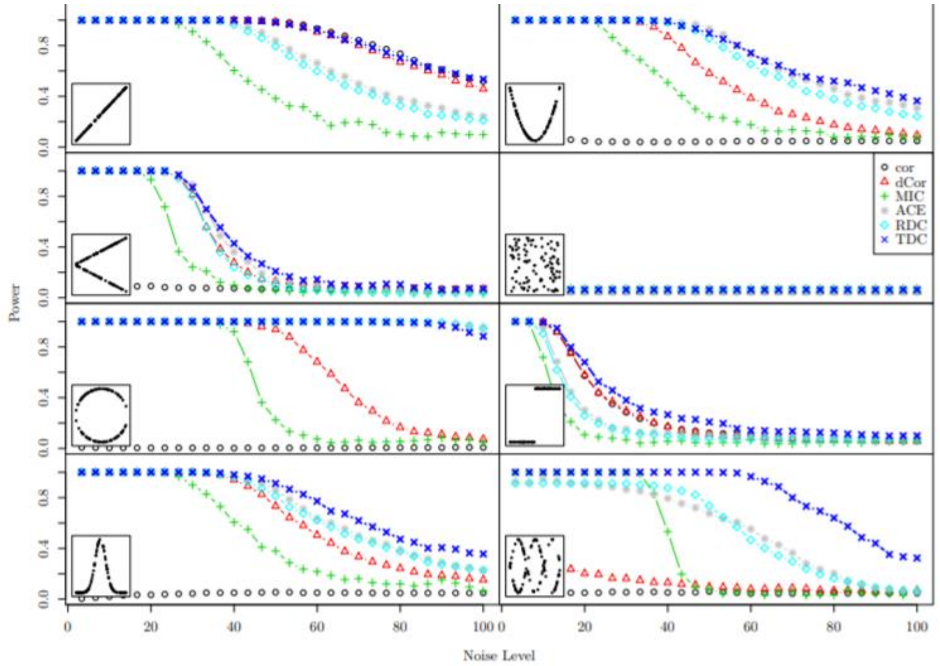
Spearman correlation ρ_S more resilient to outliers than Pearson correlation ρ

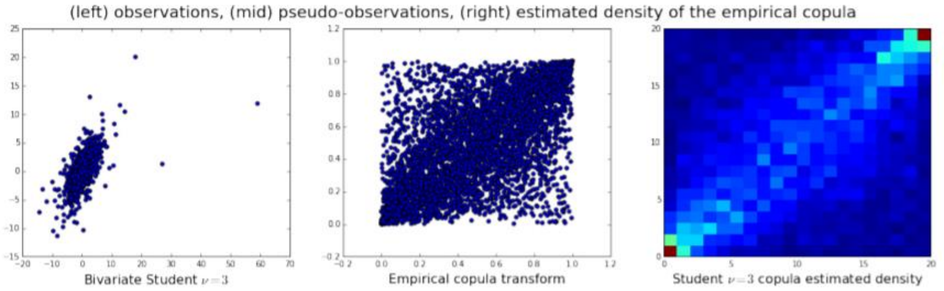


(a) $\rho = 0.66$ (b) $\rho = 0.23$ (c) $\rho_S = 0.65$ (d) $\rho_S = 0.64$

Sklar's theorem: $F(x_i, x_j) = C_{ij}(F_i(x_i), F_j(x_j))$

Copulas C = encode dependence between marginals F





- Build the forget-dependence copulas $\{C_l^F\}_l$
- Build the target-dependence copulas $\{C_k^T\}_k$
- Compute the empirical copula C_{ij} from x_i, x_j

$$\mathrm{TDC}(C_{ij}) = \frac{\min_l \mathrm{EMD}(C_l^F, C_{ij})}{\min_l \mathrm{EMD}(C_l^F, C_{ij}) + \min_k \mathrm{EMD}(C_{ij}, C_k^T)}$$

Riemannian minimum enclosing ball

$a\#_t^M b$: point $\gamma(t)$ on the geodesic line segment $[ab]$ wrt M.

Algorithm GeoA

$c_1 \leftarrow$ choose randomly a point in $\mathcal{P}$;

for $i = 2$ **to** l **do**

```
// farthest point from c_i
```

$s_i \leftarrow \arg\max_{j=1}^{n} \rho(c_i, p_j)$;

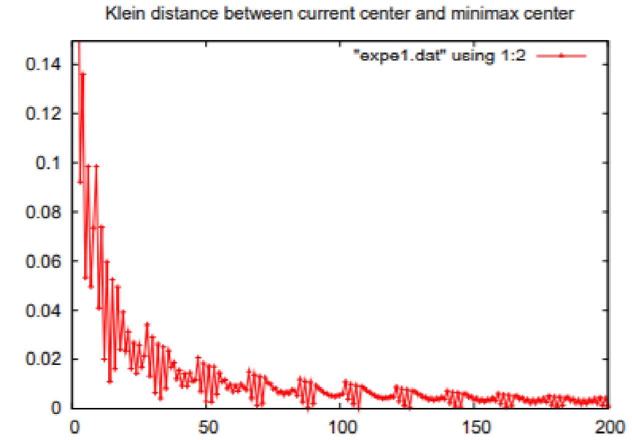
```
// update the center: walk on the geodesic line
   segment [c_i, p_{s_i}]
```

$c_{i+1} \leftarrow c_i \#^M_{\frac{1}{i+1}} p_{s_i}$;

end

```
// Return the SEB approximation
```

return $\mathrm{Ball}(c_l, r_l = \rho(c_l, \mathcal{P}))$;



Hyperbolic geometry:

$$\rho(p,q) = \operatorname{arccosh} \frac{1 - p^\top q}{\sqrt{(1 - p^\top p)(1 - q^\top q)}}$$

$$T_p\left(T_{-p}(p) \#_\alpha T_{-p}(q)\right) = p \#_\alpha q.$$

$$T_p(x) = \frac{\left(1 - \|p\|^2\right) x + \left(\|x\|^2 + 2\langle x, p\rangle + 1\right) p}{\|p\|^2 \|x\|^2 + 2\langle x, p\rangle + 1}$$

Positive-definite matrices:

$$\rho(P,Q) = \|\log(P^{-1}Q)\|_F = \sqrt{\textstyle\sum_i \log^2 \lambda_i}$$

$$\gamma_t(P,Q) = P^{\frac{1}{2}} \left(P^{-\frac{1}{2}} Q P^{-\frac{1}{2}}\right)^t P^{\frac{1}{2}}$$

@FrnkNlsn

On Approximating the Riemannian 1-Center, Comp. Geom. 2013

Approximating Covering and Minimum Enclosing Balls in Hyperbolic Geometry, GSI, 2015

Neuromanifolds, Occam's Razor and Deep Learning

Question: Why do DNNs generalize well with huge number of free parameters?

Problem: Generalization error of DNNs is experimentally
not U-shaped but a double descent risk curve (arxiv 1812.11118)

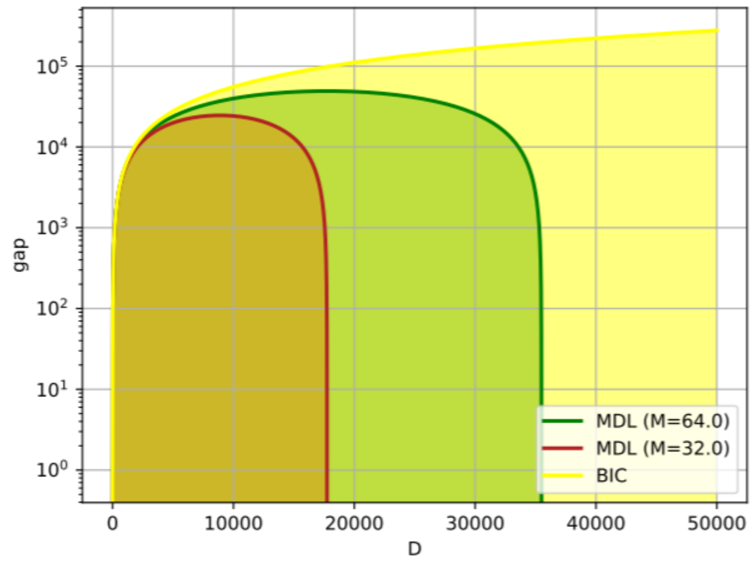
Occam's razor for Deep Neural Networks (DNNs):

(uniform width M, L layers, N #observations, d: dimension of screen distributions in lightlike neuromanifold)

Θ: parameters of the DNN, $\hat{\Theta}$: estimated parameters

$$\mathcal{O} = -\log P(X \mid \hat{\Theta}) + \frac{d}{2}\log N + \frac{d}{2}\int_0^\infty \rho_{\mathcal{I}}(\lambda)\log\lambda d\lambda$$

$$\mathcal{O} \approx -\log P(X \mid \hat{\Theta}) + \frac{d}{2}\log N - \frac{d}{2}\gamma LM$$



Estimated generalisation gap (in log scale) against the number of free parameters.

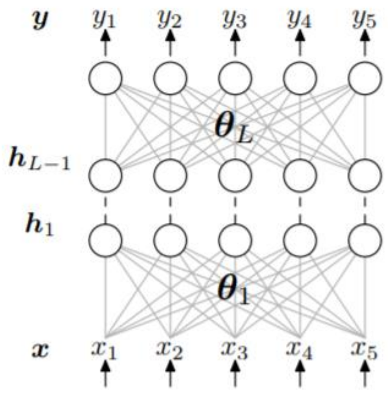
https://arxiv.org/abs/1905.11027

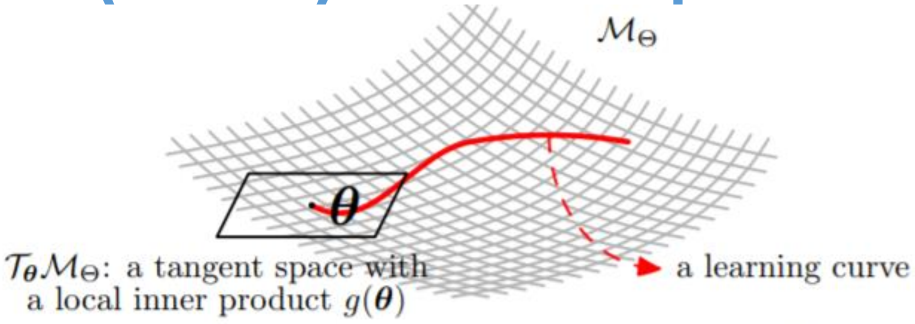
$\rho_{\mathcal{I}}$ Spectrum density of the Fisher Information Matrix (FIM)

$$\mathcal{I}(\Theta) = E_p\left(\frac{\partial \log p(X \mid \Theta)}{\partial \Theta}\frac{\partial \log p(X \mid \Theta)}{\partial \Theta^\top}\right)$$

Relative Fisher Information Matrix (RFIM) and Relative Natural Gradient (RNG) for deep learning

$$p(\boldsymbol{y} \mid \boldsymbol{x}, \Theta) = \sum_{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{L-1}} p(\boldsymbol{y} \mid \boldsymbol{h}_{L-1}, \theta_L) \cdots p(\boldsymbol{h}_2 \mid \boldsymbol{h}_1, \theta_2) p(\boldsymbol{h}_1 \mid \boldsymbol{x}, \theta_1),$$



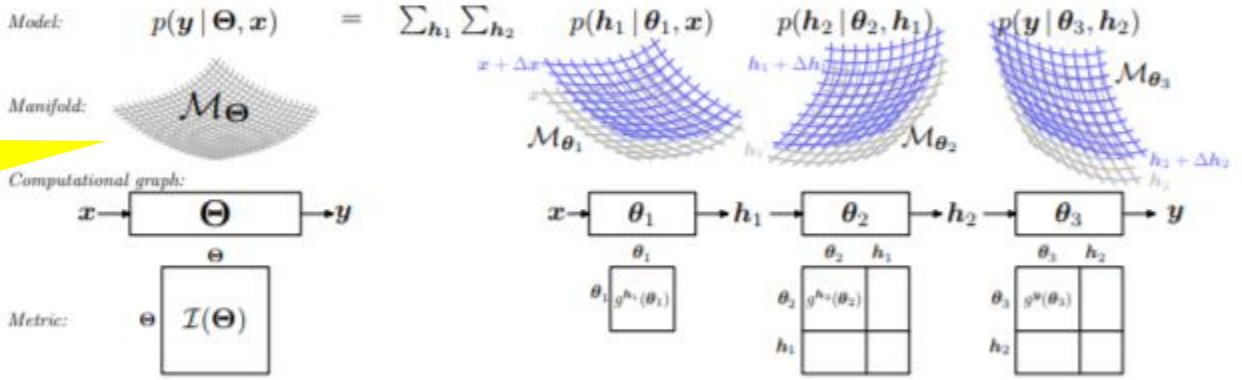


$$g(\Theta) = E_{\boldsymbol{x} \sim \hat{p}(X_n),\, \boldsymbol{y} \sim p(\boldsymbol{y} \mid \boldsymbol{x}, \Theta)} \left[\frac{\partial l}{\partial \Theta} \frac{\partial l}{\partial \Theta^{\intercal}} \right]$$

$$= \frac{1}{n} \sum_{i=1}^{n} E_{p(\boldsymbol{y} \mid \boldsymbol{x}_i, \Theta)} \left[\frac{\partial l_i}{\partial \Theta} \frac{\partial l_i}{\partial \Theta^{\intercal}} \right]$$

Relative Fisher IM: $g^{\boldsymbol{h}}(\boldsymbol{\theta} \mid \boldsymbol{\theta}_f) = E_{p(\boldsymbol{h} \mid \boldsymbol{\theta}, \boldsymbol{\theta}_f)} \left[\frac{\partial}{\partial \boldsymbol{\theta}} \ln p(\boldsymbol{h} \mid \boldsymbol{\theta}, \boldsymbol{\theta}_f) \frac{\partial}{\partial \boldsymbol{\theta}^{\intercal}} \ln p(\boldsymbol{h} \mid \boldsymbol{\theta}, \boldsymbol{\theta}_f) \right]$

Dynamic geometry



The RFIMs of single neuron models, a linear layer, a non-linear layer, a soft-max layer, two consecutive layers all have simple closed form solutions

Clustering with mixed α-Divergences

$$M_{\lambda,\alpha}(p:x:q) = \lambda D_\alpha(p:x) + (1-\lambda) D_\alpha(x:q) \quad \text{with} \quad D_\alpha(p:q) \doteq \sum_{i=1}^{d} \frac{4}{1-\alpha^2}\left(\frac{1-\alpha}{2}p^i + \frac{1+\alpha}{2}q^i - (p^i)^{\frac{1-\alpha}{2}}(q^i)^{\frac{1+\alpha}{2}}\right)$$

K-means (hard/flat clustering)

Algorithm 1: Mixed α-seeding; $\mathrm{MAS}(\mathcal{H}, k, \lambda, \alpha)$

Input: Weighted histogram set $\mathcal{H}$, integer $k \geq 1$, real $\lambda \in [0,1]$, real $\alpha \in \mathbb{R}$;

Let $\mathcal{C} \leftarrow h_j$ with uniform probability ;

for $i = 2, 3, \ldots, k$ **do**

Pick at random histogram $h \in \mathcal{H}$ with probability:

$$\pi_{\mathcal{H}}(h) \doteq \frac{w_h M_{\lambda,\alpha}(c_h : h : c_h)}{\sum_{y \in \mathcal{H}} w_y M_{\lambda,\alpha}(c_y : y : c_y)},$$

//where $(c_h, c_h) \doteq \arg\min_{(z,z)\in\mathcal{C}} M_{\lambda,\alpha}(z : h : z)$;

$\mathcal{C} \leftarrow \mathcal{C} \cup \{(h,h)\}$;

Output: Set of initial cluster centers $\mathcal{C}$;

Input: Weighted histogram set $\mathcal{H}$, integer $k > 0$, real $\lambda \in [0,1]$, real $\alpha \in \mathbb{R}$;

Let $\mathcal{C} = \{(l_i, r_i)\}_{i=1}^{k} \leftarrow \mathrm{MAS}(\mathcal{H}, k, \lambda, \alpha)$;

repeat

//Assignment

for $i = 1, 2, \ldots, k$ **do**

$\mathcal{A}_i \leftarrow \{h \in \mathcal{H} : i = \arg\min_j M_{\lambda,\alpha}(l_j : h : r_j)\}$;

// Centroid relocation

for $i = 1, 2, \ldots, k$ **do**

$r_i \leftarrow \left(\sum_{h \in \mathcal{A}_i} w_i h^{\frac{1-\alpha}{2}}\right)^{\frac{2}{1-\alpha}}$;

$l_i \leftarrow \left(\sum_{h \in \mathcal{A}_i} w_i h^{\frac{1+\alpha}{2}}\right)^{\frac{2}{1+\alpha}}$;

until *convergence*;

Output: Partition of $\mathcal{H}$ in k clusters following $\mathcal{C}$;

$$J_\alpha(\tilde{p} : \tilde{q}) = \frac{8}{1-\alpha^2}\left(1 + \sum_{i=1}^{d} H_{\frac{1-\alpha}{2}}(\tilde{p}^i, \tilde{q}^i)\right)$$

$$H_\beta(a,b) = \frac{a^\beta b^{1-\beta} + a^{1-\beta} b^\beta}{2}$$

Heinz means interpolate the arithmetic and the geometric means

$$\sqrt{ab} = H_{\frac{1}{2}}(a,b) \leq H_\alpha(a,b) \leq H_0(a,b) = \frac{a+b}{2}$$

EM (soft/generative clustering)

Input: Histogram set $\mathcal{H}$ with $|\mathcal{H}| = m$, integer $k > 0$, real $\lambda \leftarrow \lambda_{\mathrm{init}} \in [0,1]$, real $\alpha \in \mathbb{R}$;

Let $\mathcal{C} = \{(l_i, r_i)\}_{i=1}^{k} \leftarrow \mathrm{MAS}(\mathcal{H}, k, \lambda, \alpha)$;

repeat

//Expectation

for $i = 1, 2, \ldots, m$ **do**

for $j = 1, 2, \ldots, k$ **do**

$p(j|h_i) = \frac{\pi_j \exp(-M_{\lambda,\alpha}(l_j : h_i : r_j))}{\sum_{j'} \pi_{j'} \exp(-M_{\lambda,\alpha}(l_{j'} : h_i : r_{j'}))}$;

//Maximization

for $j = 1, 2, \ldots, k$ **do**

$\pi_j \leftarrow \frac{1}{m}\sum_i p(j|h_i)$;

$l_i \leftarrow \left(\frac{1}{\sum_i p(j|h_i)} \sum_i p(j|h_i) h_i^{\frac{1+\alpha}{2}}\right)^{\frac{2}{1+\alpha}}$;

$r_i \leftarrow \left(\frac{1}{\sum_i p(j|h_i)} \sum_i p(j|h_i) h_i^{\frac{1-\alpha}{2}}\right)^{\frac{2}{1-\alpha}}$;

//Alpha - Lambda

$\alpha \leftarrow \alpha - \eta_1 \sum_{j=1}^{k} \sum_{i=1}^{m} p(j|h_i) \frac{\partial}{\partial\alpha} M_{\lambda,\alpha}(l_j : h_i : r_j)$;

if $\lambda_{\mathrm{init}} \neq 0, 1$ **then**

$\lambda \leftarrow \lambda - \eta_2 \left(\sum_{j=1}^{k}\sum_{i=1}^{m} p(j|h_i) D_\alpha(l_j : h_i) - \sum_{j=1}^{k}\sum_{i=1}^{m} p(j|h_i) D_\alpha(h_i : r_j)\right)$;

//for some small η_1, η_2; ensure that $\lambda \in [0,1]$.

until *convergence*;

Output: Soft clustering of $\mathcal{H}$ according to k densities $p(j|.)$ following $\mathcal{C}$;

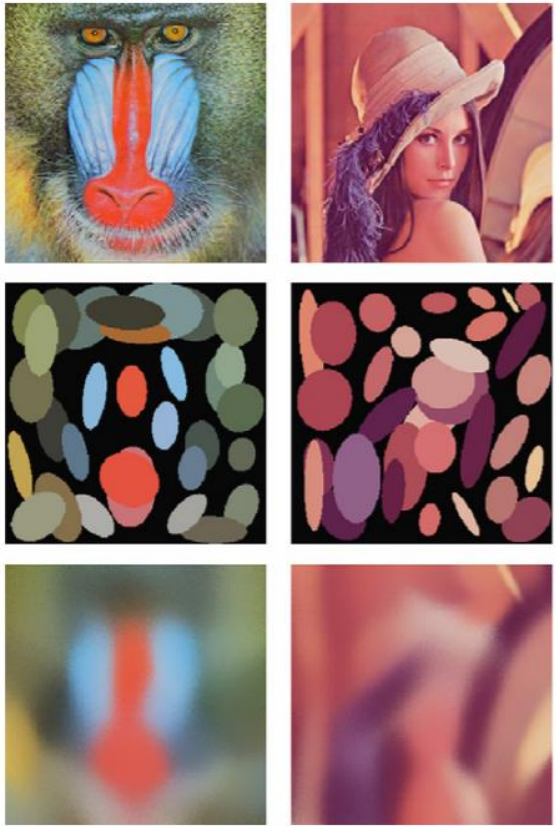
@FrnkNlsn

On Clustering Histograms with k-Means by Using Mixed α-Divergences. Entropy 16(6): 3273-3301 (2014)

Hierarchical mixtures of exponential families

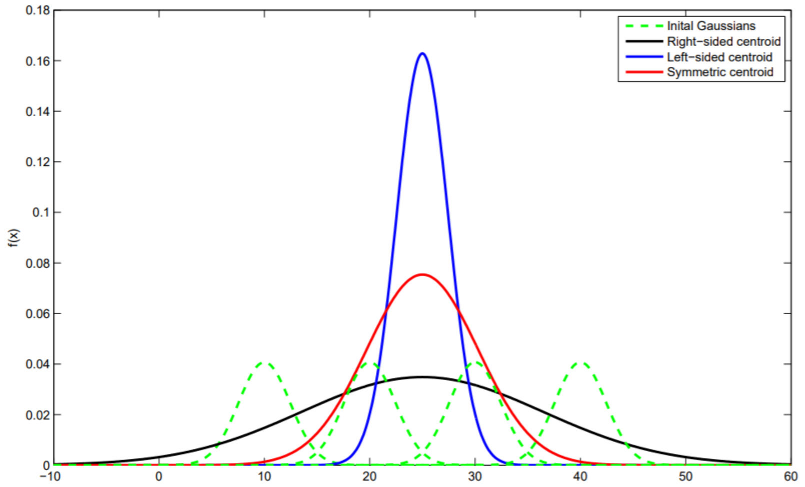
Hierarchical clustering with Bregman sided and symmetrized divergences

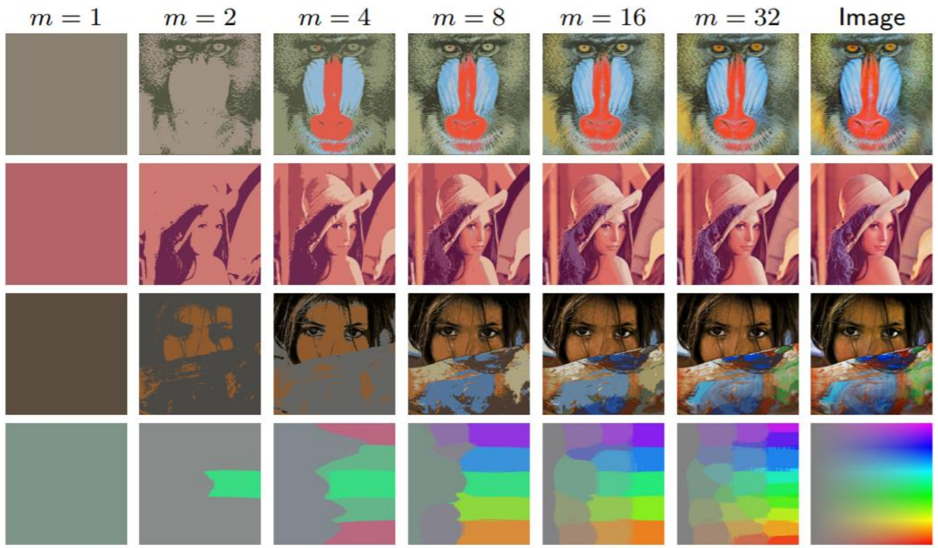
Learning & simplifying
Gaussian mixture models (GMMs)

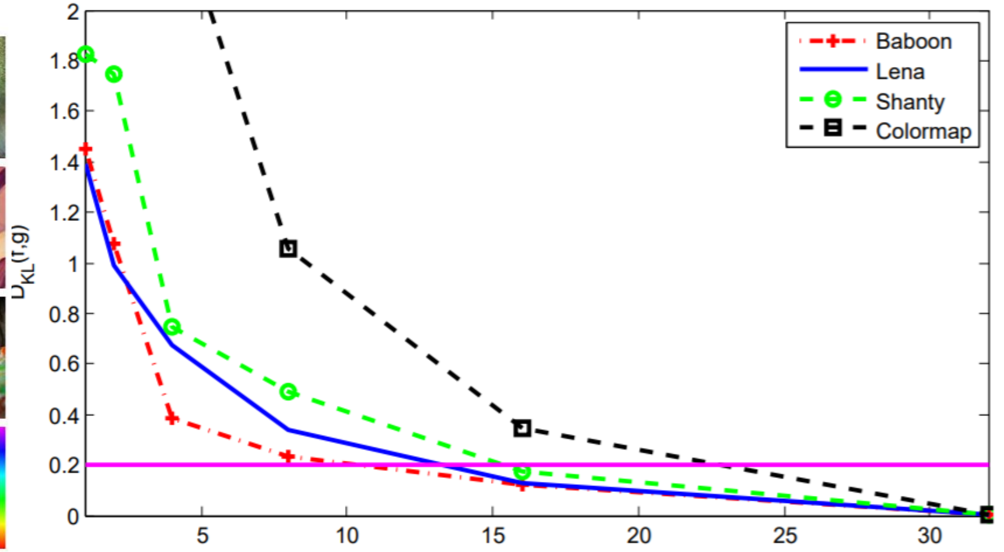


- Agglomerative method:
  1. Find the two closest subsets $\mathcal{S}_i$ and $\mathcal{S}_j$
  2. Merge the subsets $\mathcal{S}_i$ and $\mathcal{S}_j$
  3. Go back to 1. until one single set remains

Criterion	Formula
Minimum distance	$D_{\min}(A,B) = \min\{d(a,b) \mid a \in A,\ b \in B\}$
Maximum distance	$D_{\max}(A,B) = \max\{d(a,b) \mid a \in A,\ b \in B\}$
Average distance	$D_{\mathrm{av}}(A,B) = \frac{1}{\lvert A \rvert \lvert B \rvert} \sum_{a\in A} \sum_{b \in B} d(a,b)$



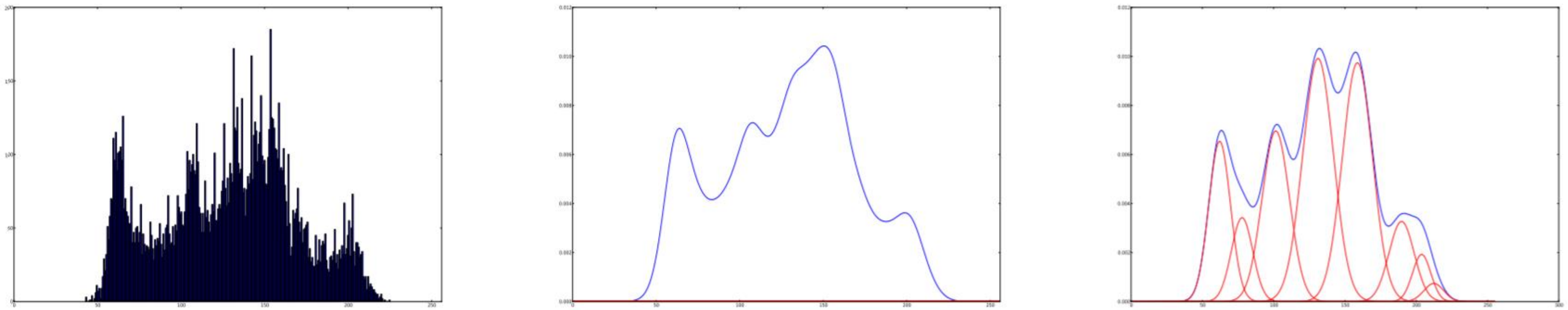




@FrnkNlsn

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

Learning a mixture by simplifying a kernel density estimator



Original histogram
raw KDE (14400 components)
simplified mixture (8 components)

$$f(x) = \sum_{i=1}^{n} \omega_i \, g(x;\, \mu_i, \sigma_i^2) \qquad g(x;\, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Usual **centroids** based on Kullback-Leibler sided/symmetrized

$$\arg\min_c \textstyle\sum_i \omega_i KLD(c, x_i)$$

$$\arg\min_c \textstyle\sum_i \omega_i KLD(x_i, c)$$

$$\arg\min_c \textstyle\sum_i \omega_i \mathrm{SKL}(x_i, c)$$

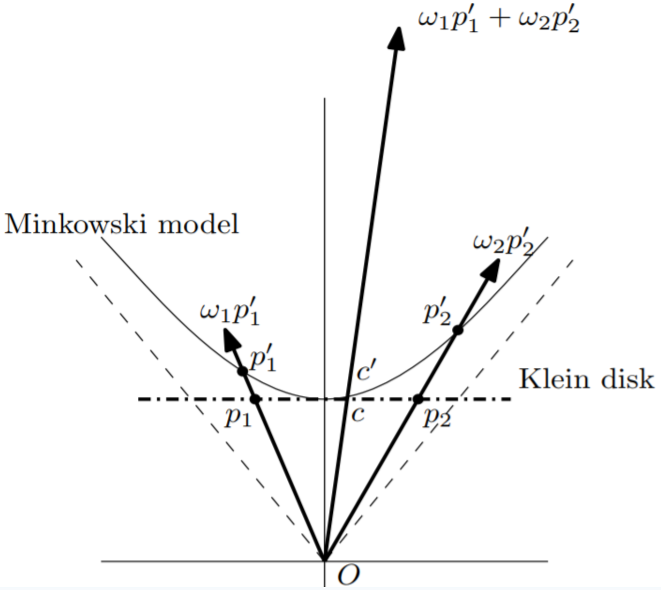
$$\mathrm{KLD}(f_p, f_q) = \frac{1}{2}\log\left(\frac{\det \Sigma_p}{\det \Sigma_q}\right) + \frac{1}{2}\mathrm{tr}(\Sigma_q^{-1}\Sigma_p) + \frac{1}{2}(\mu_q - \mu_p)^T \Sigma_q^{-1}(\mu_q - \mu_p) - \frac{d}{2}$$

or Fisher-Rao distance (hyperbolic distance)

$$\mathrm{FRD}(f_p, f_q) = \sqrt{2}\ln \frac{|(\frac{\mu_p}{\sqrt{2}}, \sigma_p) - (\frac{\mu_q}{\sqrt{2}}, \sigma_q)| + |(\frac{\mu_p}{\sqrt{2}}, \sigma_p) - (\frac{\mu_q}{\sqrt{2}}, \sigma_q)|}{|(\frac{\mu_p}{\sqrt{2}}, \sigma_p) - (\frac{\mu_q}{\sqrt{2}}, \sigma_q)| - |(\frac{\mu_p}{\sqrt{2}}, \sigma_p) - (\frac{\mu_q}{\sqrt{2}}, \sigma_q)|}$$

Pb: No closed-form FR/SKL centroids!!!

Galperin's model centroid (HG)



Simple model centroid algorithm:
Embed Klein points to points of the Minkowski hyperboloid
Centroid = center of mass c, scaled back to c' of the hyperboloid
Map back c' to Klein disk

@FrnkNlsn

Model centroids for the simplification of Kernel Density estimators. ICASSP 2012

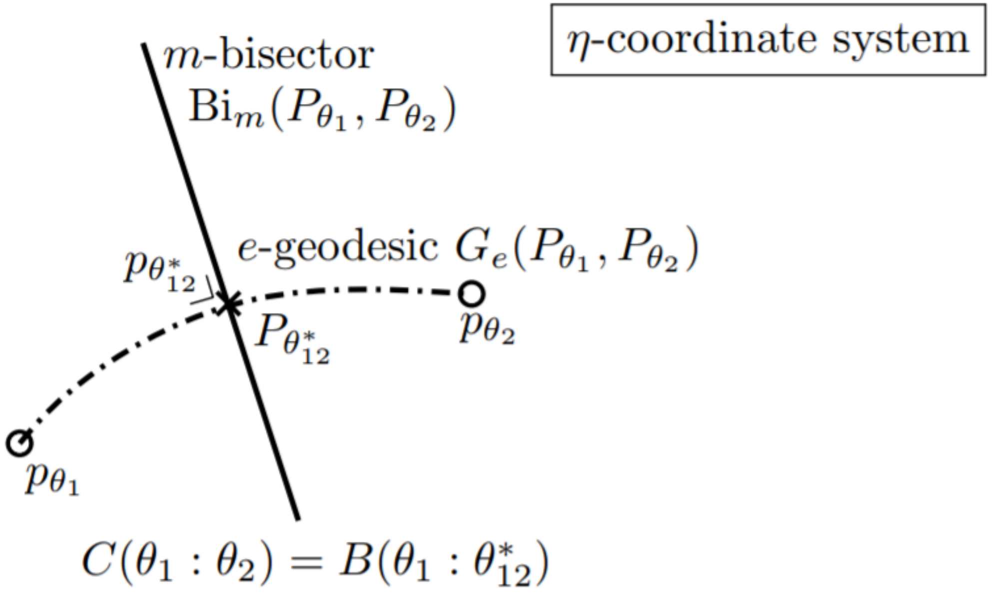
Bayesian hypothesis testing:
A geometric characterization of the best error exponent

Dually flat Exponential Family Manifold (EFM):
Chernoff information amounts to a Bregman divergence

$$P^* = P_{\theta_{12}^*} = G_e(P_1, P_2) \cap \mathrm{Bi}_m(P_1, P_2)$$

$$C(P_1, P_2) = -\log \min_{\alpha \in (0,1)} \int_{x \in \mathcal{X}} p_1^{\alpha}(x) p_2^{1-\alpha}(x) \mathrm{d}\nu(x) \geq 0,$$

Chernoff Information



This geometric characterization yields to an exact closed-form solution in 1D EFs, and a simple geodesic bisection search for arbitrary dimension

@FrnkNlsn

An Information-Geometric Characterization of Chernoff Information, IEEE SPL, 2013 (arXiv:1102.2684)

Muti-continued fractions

$$\frac{p}{q} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\ddots + \cfrac{1}{a_n}}}}}$$

n-th convergent	Continued Fraction Expansion	Rational	Decimal Rep.
0	[1]	$\frac{1}{1}$	1
1	[1/2]	$\frac{3}{2}$	1.5
2	[1/2/3]	$\frac{10}{7}$	1.428571429...
3	[1/2/3/4]	$\frac{43}{30}$	1.433333333...
4	[1/2/3/4/5]	$\frac{225}{157}$	1.433121019...

Convergents of $\frac{225}{157} = [1/2/3/4/5]$.

1.000000000	$\frac{1}{1}$	$1/\infty$
1.500000000	$\frac{3}{2}$	$1/2/\infty$
1.570000000	$\frac{157}{100}$	$1/1/1/3/14/\infty$
1.570000000	$\frac{157}{100}$	$1/1/1/3/14/\infty$
1.570700000	$\frac{15707}{10000}$	$1/1/1/3/27/1/2/1/1/1/4/\infty$
1.570790000	$\frac{157079}{100000}$	$1/1/1/3/31/1/2/16/4/2/\infty$
1.570796000	$\frac{392699}{250000}$	$1/1/1/3/31/1/41/1/1/2/1/3/\infty$
1.570796300	$\frac{15707963}{10000000}$	$1/1/1/3/31/1/121/3/1/4/3/1/1/2/\infty$
1.570796320	$\frac{9817477}{6250000}$	$1/1/1/3/31/1/138/1/2/2/1/4/4/\infty$
1.570796326	$\frac{785398163}{500000000}$	$1/1/1/3/31/1/144/1/18/8/2/1/7/4/\infty$

Approximations of $\frac{\pi}{2}$.

Matrix representation of continued fractions

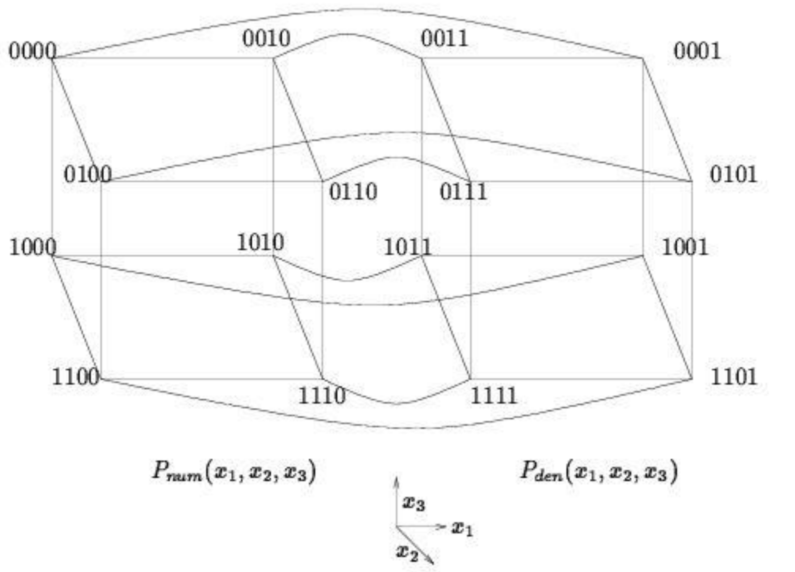
$$\frac{p}{q} = [a_0/\ldots/a_n]$$

If n is odd:

$$\frac{p}{q} \equiv (p\ q) = (x_i\ 1) = (1\ 0) \times \begin{pmatrix} 1 & a_n \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ a_{n-1} & 1 \end{pmatrix} \cdots \begin{pmatrix} 1 & 0 \\ a_{2j} & 0 \end{pmatrix} \begin{pmatrix} 1 & a_{2j-1} \\ 0 & 1 \end{pmatrix} \cdots \begin{pmatrix} 1 & 0 \\ a_0 & 1 \end{pmatrix}$$

If n is even:

$$\frac{p}{q} \equiv (p\ q) = (x_i\ 1) = (0\ 1) \times \begin{pmatrix} 1 & 0 \\ a_n & 1 \end{pmatrix} \begin{pmatrix} 1 & a_{n-1} \\ 0 & 1 \end{pmatrix} \cdots \begin{pmatrix} 1 & a_{2j+1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ a_{2j} & 0 \end{pmatrix} \cdots \begin{pmatrix} 1 & 0 \\ a_0 & 1 \end{pmatrix}$$

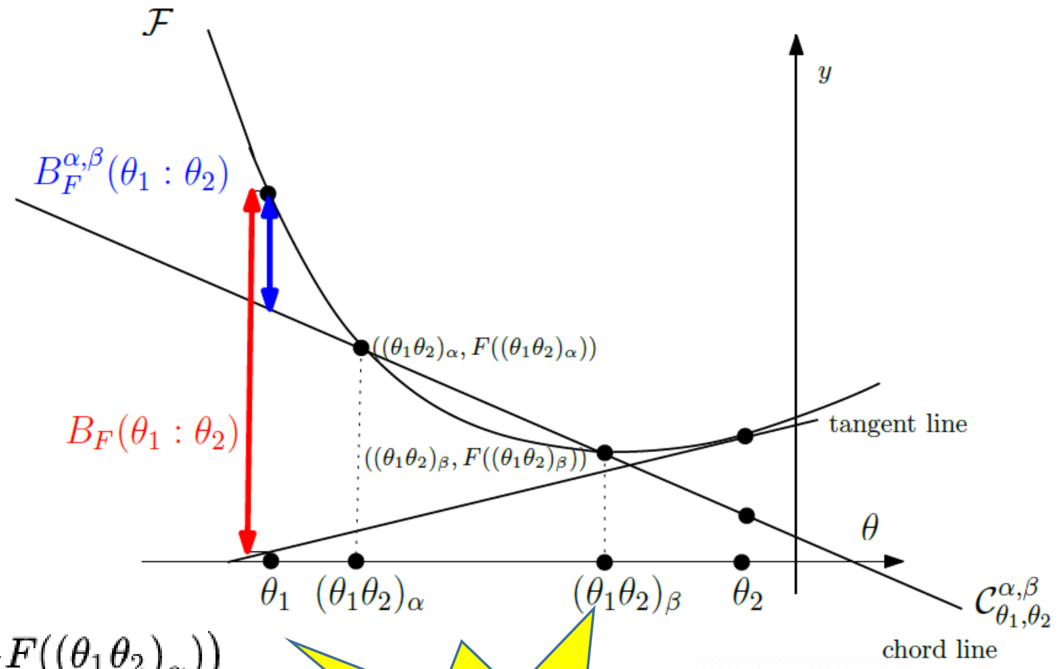


Algorithms on Continued and Multi-continued fractions, 1993

Bregman chord divergence: Free of gradient!

Ordinary Bregman divergence requires gradient calculation:

$$B_F(\theta_1 : \theta_2) = F(\theta_1) - F(\theta_2) - (\theta_1 - \theta_2)^\top \nabla F(\theta_2)$$

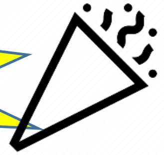


Bregman chord divergence

uses two extra scalars α and β:

$$B_F^{\alpha,\beta}(\theta_1 : \theta_2) = F(\theta_1) - F\left((\theta_1\theta_2)_\alpha\right) - \frac{\alpha\left(F((\theta_1\theta_2)_\beta) - F((\theta_1\theta_2)_\alpha)\right)}{\beta-\alpha}$$

No gradient!



Using linear interpolation notation $(\theta_1\theta_2)_\alpha = (1-\alpha)\theta_1 + \alpha\theta_2$

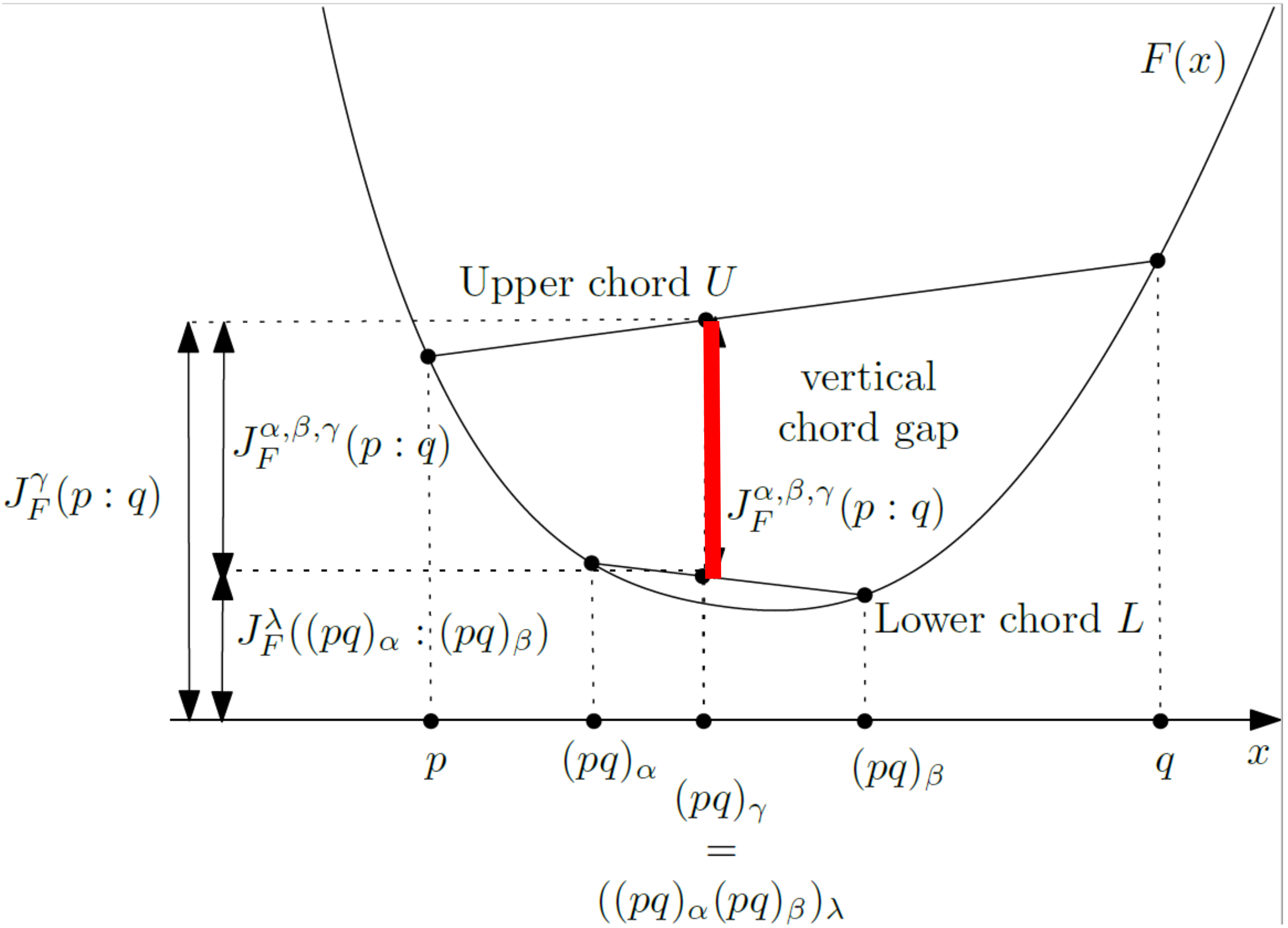
$\lim_{\beta\to\alpha} B_F^{\alpha,\beta}(\theta_1 : \theta_2) = B_F^{\alpha}(\theta_1 : \theta_2)$ and $B_F(\theta_1 : \theta_2) \simeq_{\epsilon\to 0} B_F^{1-\epsilon,1}(\theta_1 : \theta_2)$

Subfamily of **Bregman tangent divergences**: $B_F^{\alpha}(\theta_1 : \theta_2) = F(\theta_1) - F\left((\theta_1\theta_2)_\alpha\right) - \alpha(\theta_1 - \theta_2)^\top \nabla F\left((\theta_1\theta_2)_\alpha\right)$

@FrnkNlsn

The Bregman chord divergence, arXiv:1810.09113

The Jensen chord divergence: Truncated skew Jensen divergences



Linear interpolation (LERP):

$$(pq)_\lambda := (1-\lambda)p + \lambda q$$

$$J_F^{\alpha,\beta,\gamma}(p:q) = (F(p)F(q))_\gamma - (F((pq)_\alpha)F((pq)_\beta))_\lambda$$

$$((pq)_\alpha(pq)_\beta)_\lambda = (pq)_\gamma \text{ with } \gamma \in (\alpha,\beta)$$

$$\boxed{J_F^{\alpha,\beta,\gamma}(p:q) = (F(p)F(q))_\gamma - (F((pq)_\alpha)F((pq)_\beta))_{\frac{\gamma-\alpha}{\beta-\alpha}}}$$

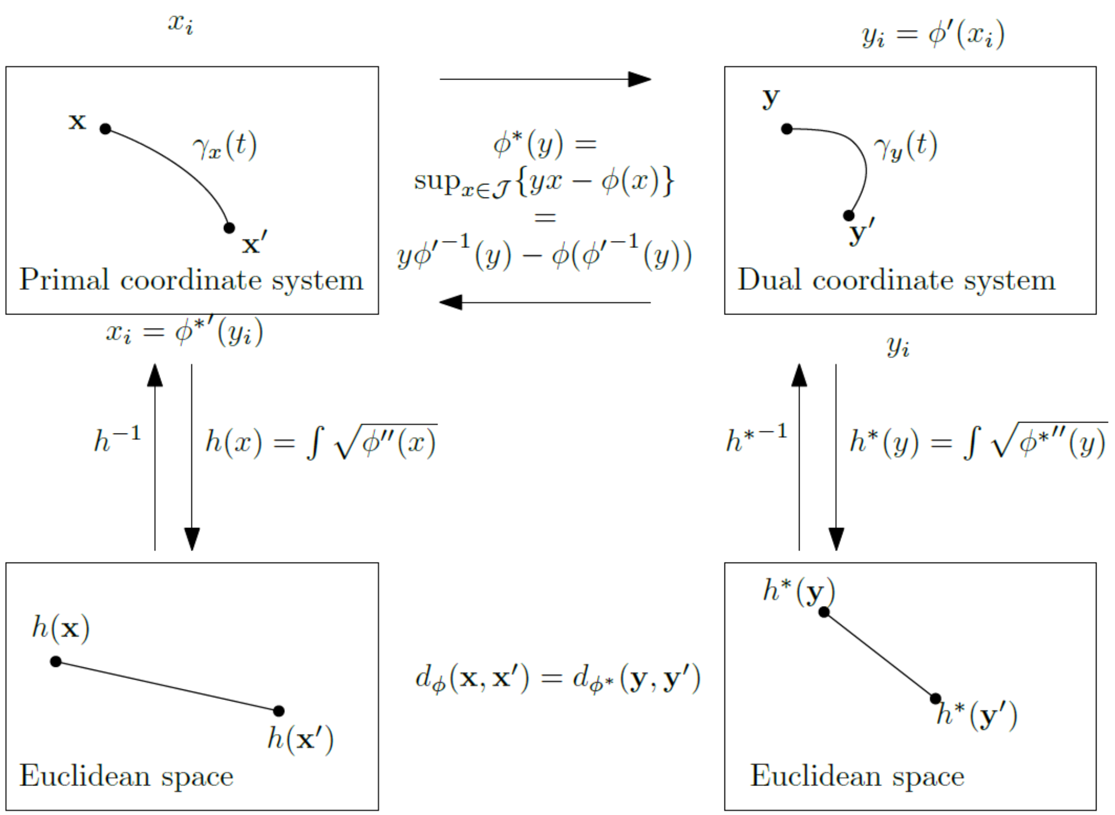
$$J_F^{\beta,\gamma}(p:q) = \gamma\left(\left(\frac{1}{\beta}-1\right)F(p) + F(q) - \frac{1}{\beta}F((pq)_\beta)\right)$$

A property:

(truncated skew Jensen divergence)

$$J_F^{\alpha,\beta,\gamma}(p:q) = J_F^{\gamma}(p:q) - J_F^{\lambda}((pq)_\alpha:(pq)_\beta)$$

Dual Riemann geodesic distances induced by a separable Bregman divergence



$$\boxed{\rho_\phi(x,x') = \rho_{\phi^*}(y,y') = \rho_{\phi^*}(\nabla\Phi(x),\nabla\Phi(x'))}$$

Legendre conjugate: $\phi^*(y) = y\phi'^{-1}(y) - \phi(\phi'^{-1}(y))$

Bregman divergence:

$$B_\Phi(x,x') := \Phi(x) - \Phi(x') - (x-x')^\top\nabla\Phi(x')$$

Separable Bregman generator:

$$\Phi(x) := \sum_{j=1}^K \phi(x_j) \text{ with } \phi : \mathcal{J} \to \mathbb{R}$$

Riemannian metric tensor:

$$g_{ij}(x) = \phi''(x_i)\delta_{ij}$$

Geodesics:

$$\gamma_i(t) = h^{-1}\Big((1-t)h(x_i) + th(x_i')\Big), \quad t\in[0,1].$$

Riemannian distance (metric):

$$\rho_\phi(x,x') = \sqrt{\sum_{j=1}^K \left(h(x_j) - h(x_j')\right)^2}$$

where $h(x) := \int\sqrt{\phi''(x)}$

Geometry and clustering with metrics derived from separable Bregman divergences, arXiv:1810.10770

Upper bounding the differential entropy (of mixtures)

<u>Idea</u>: compute the differential entropy of a **MaxEnt exponential family** with **given sufficient statistics** in closed form. *Any other* distribution has less entropy for the same moment expectations. Applies to statistical mixtures.

$$H(X) = \int_{\mathcal{X}} p(x) \log \frac{1}{p(x)} \mathrm{d}x = -\int_{\mathcal{X}} p(x) \log p(x) \mathrm{d}x$$

$$\exp(\langle \theta, t(x) \rangle - F(\theta) + k(x))$$

$$H(p(x;\theta)) = -F^*(\eta(\theta))$$

Legendre-Fenchel conjugate

Absolute Monomial Exponential Family (AMEF): $p_l(x;\theta) = \exp\left(\theta |x|^l - F_l(\theta)\right)$

with log-normalizer $F_l(\theta) = \log 2 + \log \Gamma\left(\frac{1}{l}\right) - \log l - \frac{1}{l} \log(-\theta)$

$\Gamma(u) = \int_0^\infty x^{u-1} \exp(-x) \mathrm{d}x$

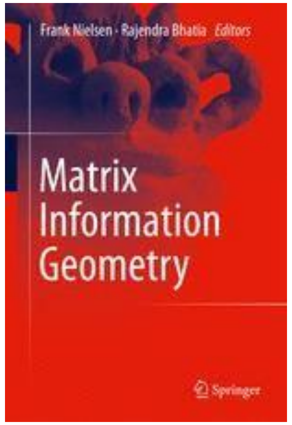
$\Gamma(n) = (n-1)!$ for $n \in \mathbb{N}$

$$H_l(\eta) = \log 2 + \log \Gamma\left(\frac{1}{l}\right) - \log l + \frac{1}{l}(1 + \log l + \log \eta)$$

@FrnkNlsn

MaxEnt Upper Bounds for the Differential Entropy of Univariate Continuous Distributions, IEEE SPL 2017, arxiv:1612.02954

Matrix Bregman divergences



For real symmetric matrices:

$$B_F(L:N) = F(L) - F(N) - \mathrm{tr}\left((L-N)\nabla_F^\top(N)\right)$$

where F is a strictly convex and differentiable generator $F : \mathrm{Sym}(d,d) \to \mathbb{R}$

- Squared Froebenius distance for $F(X) = \|X\|_F^2$
- von Neumann divergence for $F(X) = \mathrm{tr}(X\log X - X)$

$$D_{\mathrm{vN}}(X,:Y) = \mathrm{tr}(X\log X - X\log Y - X + Y)$$

- Log-det divergence for $F(X) = -\log\det(X)$

$$D_{\mathrm{ld}}(X:Y) = \mathrm{tr}\left(XY^{-1}\right) - \log\det\left(XY^{-1}\right) - n$$

Bregman–Schatten p-divergences…

Curved Mahalanobis distances (Cayley-Klein geometry)

Usual squared **Mahalanobis distance** (Bregman divergence with dually flat geometry)

$$D_Q(p,q) = \sqrt{(p-q)^\top Q(p-q)}$$

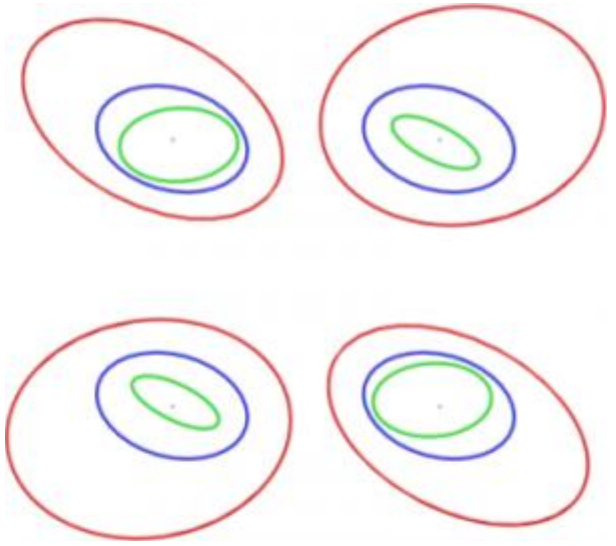
where Q is positive-definite matrix

Curved Mahalanobis distance (centered at μ and of curvature κ):

$$\boxed{D_S(p,q) = \frac{1}{2|\kappa|}\arccos\left(\frac{|S(p,q)|}{\sqrt{S(p,p)S(q,q)}}\right)}$$

$$\operatorname{arccosh}(x) = \log\left(x + \sqrt{x^2-1}\right)$$

$$S(p,q) = S_{\Sigma,\mu,\kappa}(p,q) = (p-\mu)^\top \Sigma (q-\mu) + \operatorname{sign}(\kappa)\frac{1}{\kappa^2}$$



Some curved Mahalanobis balls (Mahalanobis in blue)

@FrnkNlsn **Classification with mixtures of curved Mahalanobis metrics, ICIP 2016.**

Hölder projective divergences (incl. Cauchy-Schwarz div.)

A divergence D is **projective** when $D(\lambda p : \lambda' q) = D(p : q), \quad \forall \lambda, \lambda' > 0$

For α>0, define **conjugate exponents**: $\frac{1}{\alpha} + \frac{1}{\beta} = 1$

For α ,γ>0, define the family of **Hölder projective divergences**:

$$\boxed{D^H_{\alpha,\gamma}(p : q) = -\log\left(\frac{\int_\mathcal{X} p(x)^{\gamma/\alpha} q(x)^{\gamma/\beta} \mathrm{d}\mu(x)}{\left(\int_\mathcal{X} p(x)^\gamma \mathrm{d}\mu(x)\right)^{1/\alpha} \left(\int_\mathcal{X} q(x)^\gamma \mathrm{d}\mu(x)\right)^{1/\beta}}\right)}$$

When α=β=γ=2, we get the Cauchy-Schwarz divergence:

$$\mathrm{CS}(p : q) = -\log \frac{\int_\mathcal{X} p(x) q(x) \mathrm{d}\mu(x)}{\sqrt{\left(\int_\mathcal{X} p(x)^2 \mathrm{d}\mu(x)\right)\left(\int_\mathcal{X} q(x)^2 \mathrm{d}\mu(x)\right)}}$$

Gradient and Hessian on a manifold (M,g,∇)

Directional derivative of f at point x in direction of vector v:

$$Df(x)[v] = \lim_{t\to 0} \frac{f(x+tv)-f(x)}{t}$$

Gradient (requires metric tensor g): unique vector $\mathrm{grad}_p f(x)$ satisfying

$$\langle \mathrm{grad}_p f(x), v\rangle = Df(x)[v], \quad \forall v \in T_pM$$

Hessian (requires an affine connection, usually Levi-Civita metric conn.)

$$\mathrm{hess}_p f(x)[v] = \nabla_v^{\mathrm{LC}} \mathrm{grad}_p f(x), \quad \forall v \in T_pM$$

Property: $\langle \mathrm{hess}_p f(v)[v], w\rangle_p = \langle v, \mathrm{hess}_p f(x)[w]\rangle_p, \forall v, w \in T_pM$

@FrnkNlsn

https://arxiv.org/abs/1808.08271

Video stippling/video pointillism (CG)

Figure *We extract two density maps from the original source image: one is the color map and the other the frequency map. Then we apply our stippling algorithm and finally add both contrast and color information. We end the process by summing these two contributions — the classical (3000 points) and frequency approaches (6000 points).*

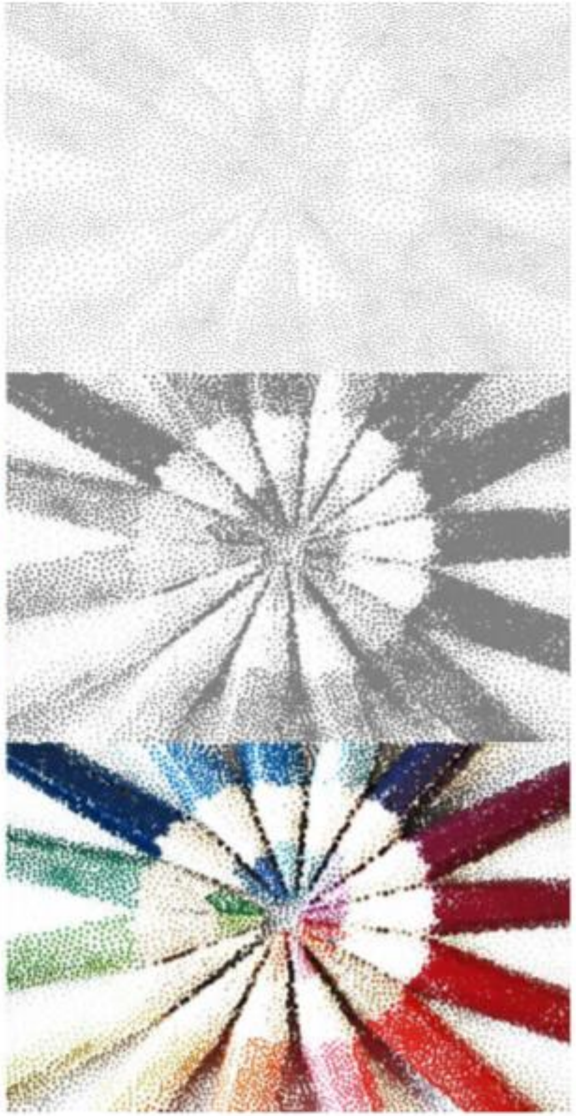


Figure *(Top) Stippling result without contrast and color. (Middle) stippling with contrast. (Bottom) stippling with color to enhance the rendering. (3000 points)*

Video

https://www.youtube.com/watch?v=O97MrPsISNk

@FrnkNlsn

Video stippling, ACIVS 2011. arXiv:1011.6049

Matching image superpixels by Earth mover distance

Superpixels by image segmentation:

- Quickshift (mean shift)
- Statistical Region Merging (SRM)

Optimal transport between superpixels including topological constraints when a **segmentation tree** is available

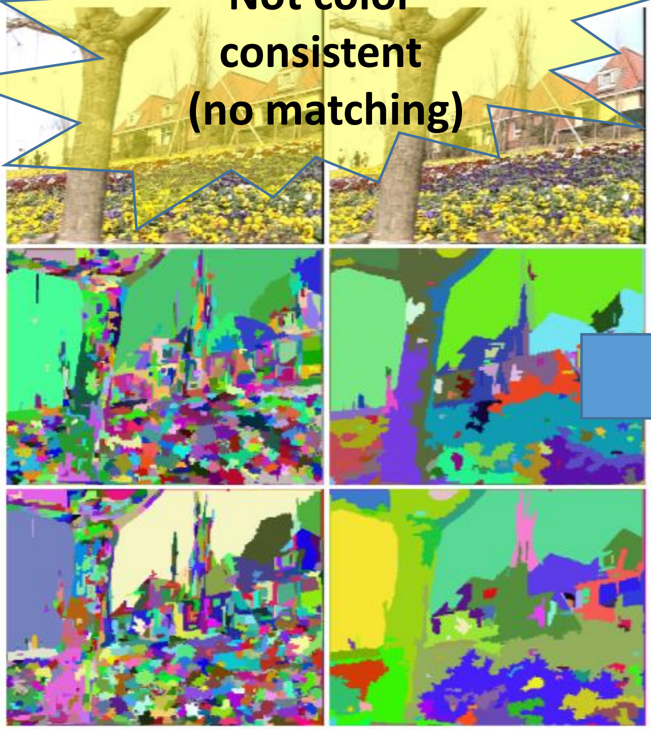


Fig. 1. Topological constraints for robust matching. From left to right, top to bottom: first image, second image, superpixelization of the first image at two different scales, superpixelization of the second image at two different scales. Toplogical constraints in EMD adds the cost of matching superpixelization at a coarse scale to the cost of matching superpixelization at a fine scale.

Fig. 2. Matching superpixelizations, From left two right, op to bottom : first image, second image, superpixelization of the first image (false color), superpixelization of the second image (false colors). Even between two images with small differences, the superpixelization, here in false colors, can be quite inconsistent. The matching of these two superpixel maps is in the color code: Superpixel i in image 2 have the color of the Superpixel in image 1 with label $j = \arg\max F(j, i)$.

@FrnkNlsn

Earth mover distance on superpixels, ICIP 2010

α-representations of the Fisher Information Matrix

Usually, the Fisher Information Matrix (FIM) is introduced in two ways:

$$I(\theta) := (I_{ij}(\theta)), \quad I_{ij}(\theta) := E_{p(x;\theta)}[\partial_i l(x;\theta)\partial_j l(x;\theta)]$$

$$I'_{ij}(\theta) := 4 \int \partial_i \sqrt{p(x;\theta)} \partial_j \sqrt{p(x;\theta)} \mathrm{d}\nu(x)$$

α-likelihood function $l^{(\alpha)}(x;\theta) := k_\alpha(p(x;\theta))$

α-Embedding

$$k_\alpha(u) = \begin{cases} \frac{2}{1-\alpha} u^{\frac{1-\alpha}{2}}, & \text{if } \alpha \neq 1 \\ \log u, & \text{if } \alpha = 1. \end{cases}$$

α-representation of the FIM:

$$\boxed{I_{ij}^{(\alpha)}(\theta) = \int \partial_i l^{(\alpha)}(x;\theta) \partial_j l^{(-\alpha)}(x;\theta) d\nu(x)}$$

Corresponds to a basis choice in the tangent space (α-base)

@FrnkNlsn

https://tinyurl.com/yyukx86o