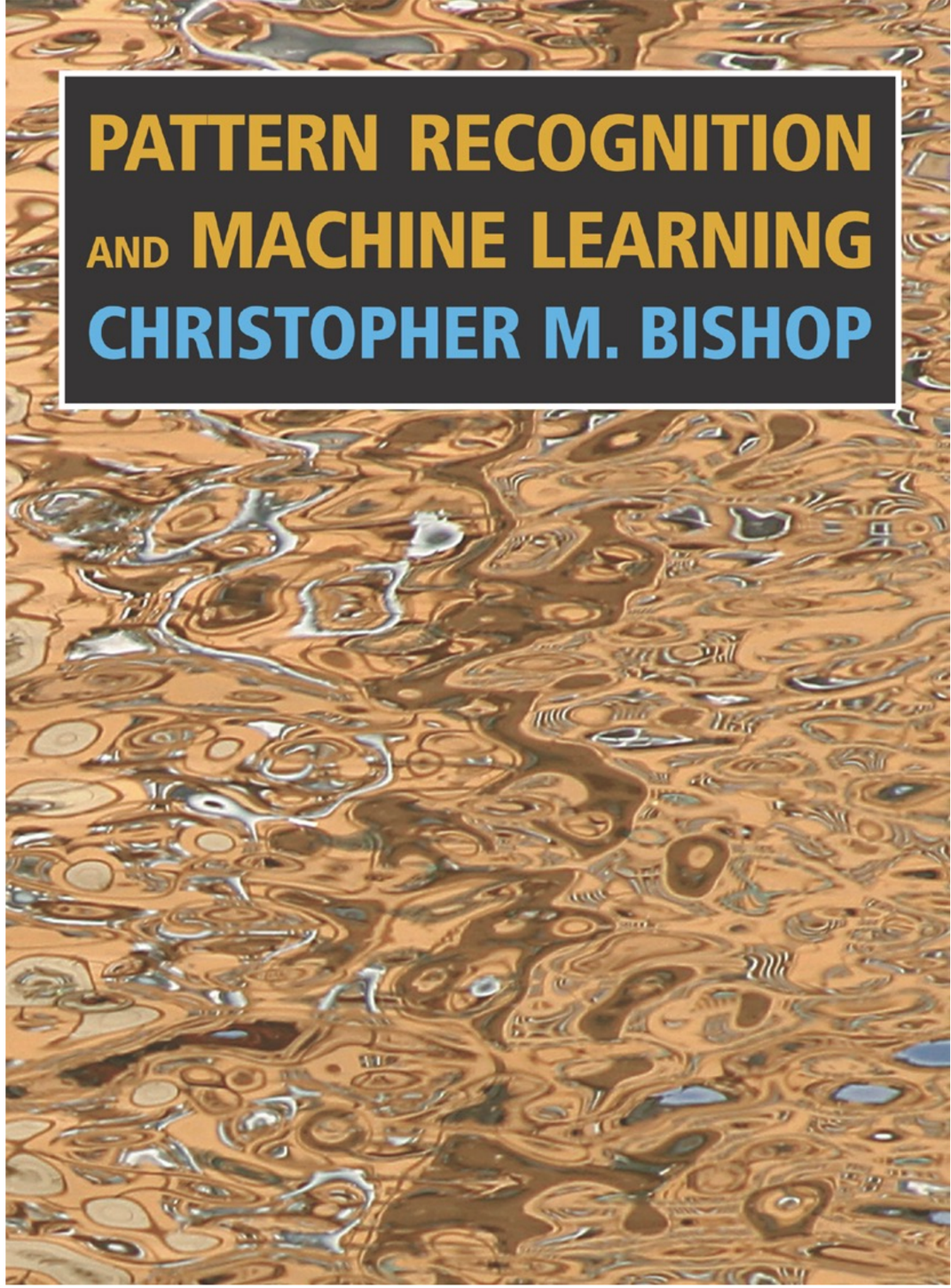
# CS 736
# Medical Image Computing (MIC)

## Image Segmentation

Suyash P. Awate

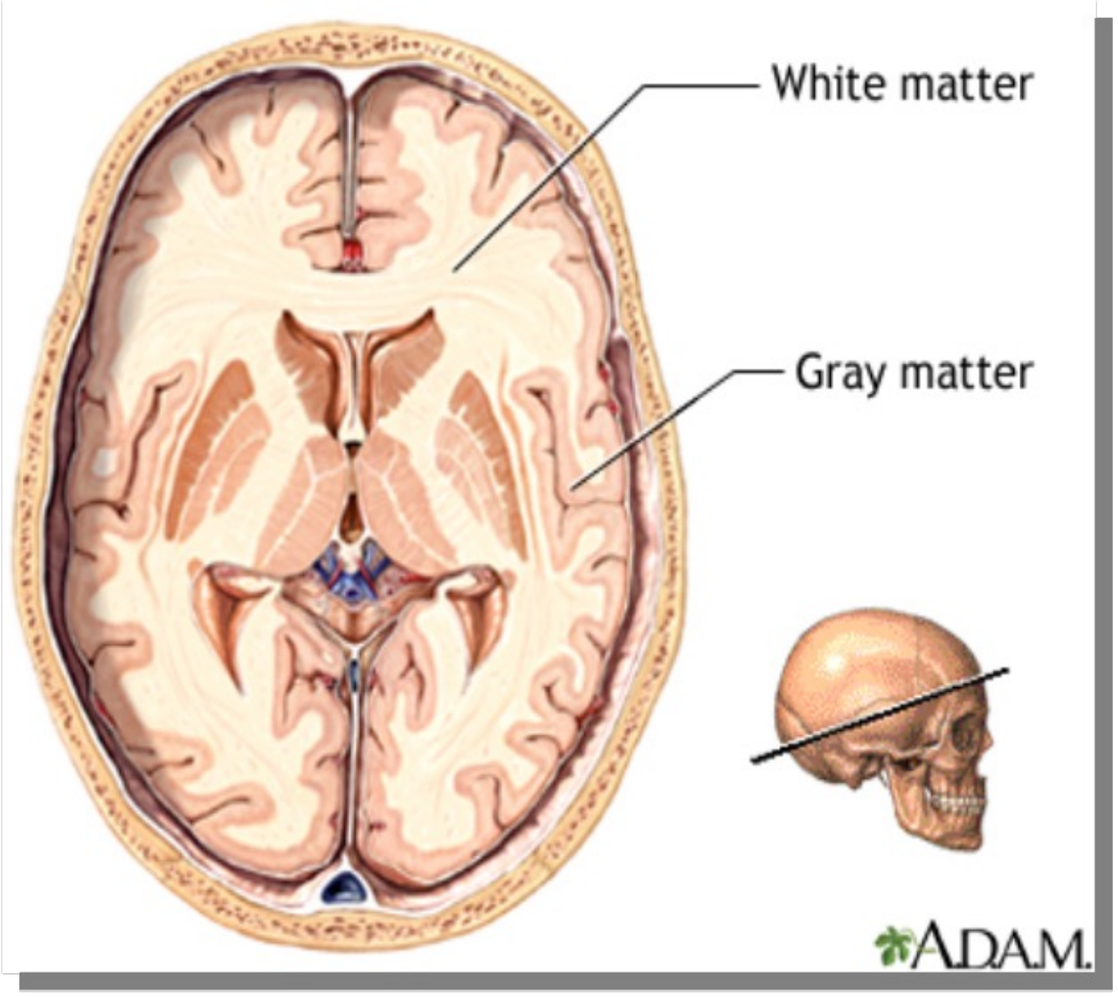
# Books

- Reference book

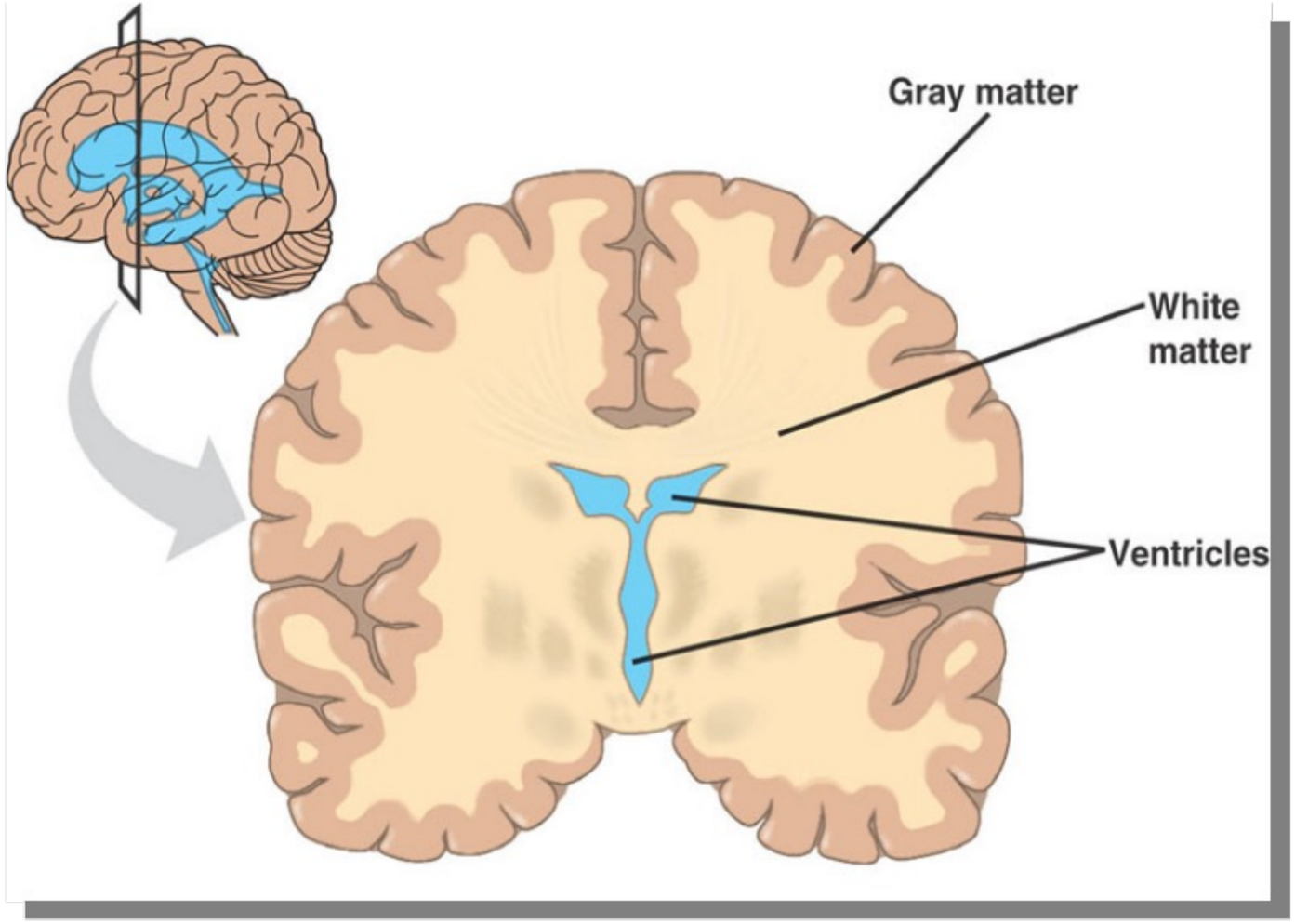


# Image Segmentation

- Labeling each voxel to a known category
- Example: human brain tissues



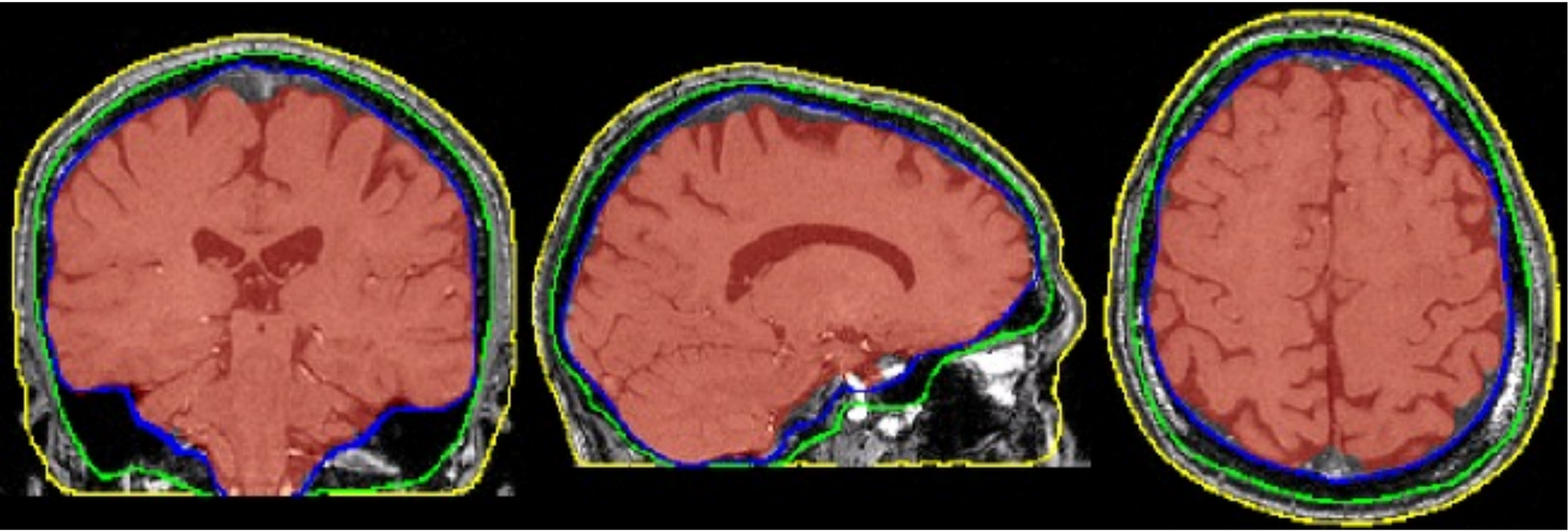

www.nlm.nih.gov/medlineplus/ency/imagepages/18117.htm




tgesicsebiology.weebly.com/nervous-system.html

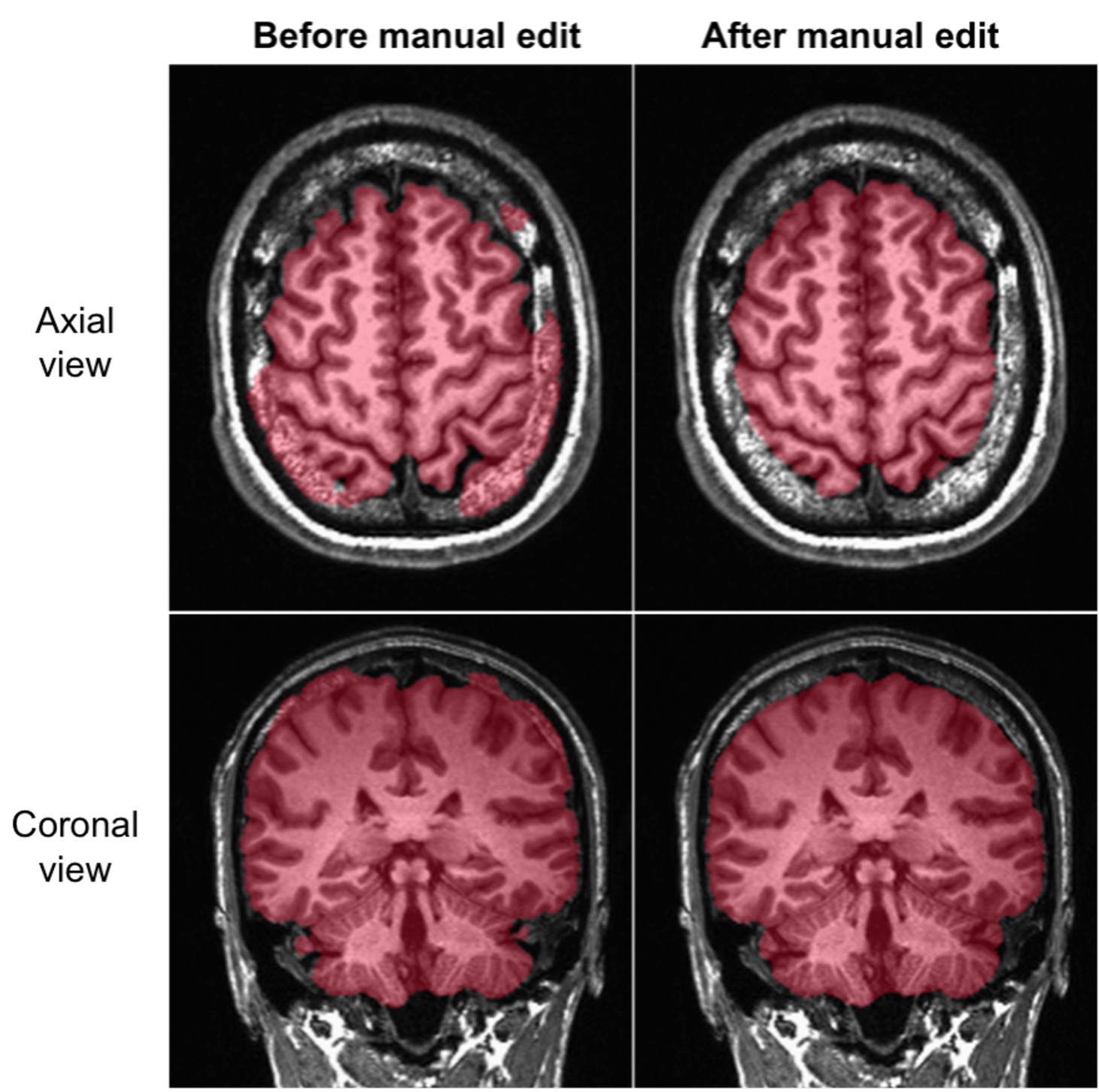
# Image Segmentation

- Goal: locate all voxels for gray matter (GM), white matter (WM), cerebrospinal fluid (CSF)
- Strategy (2 stages)
    1. Brain extraction from head MR image



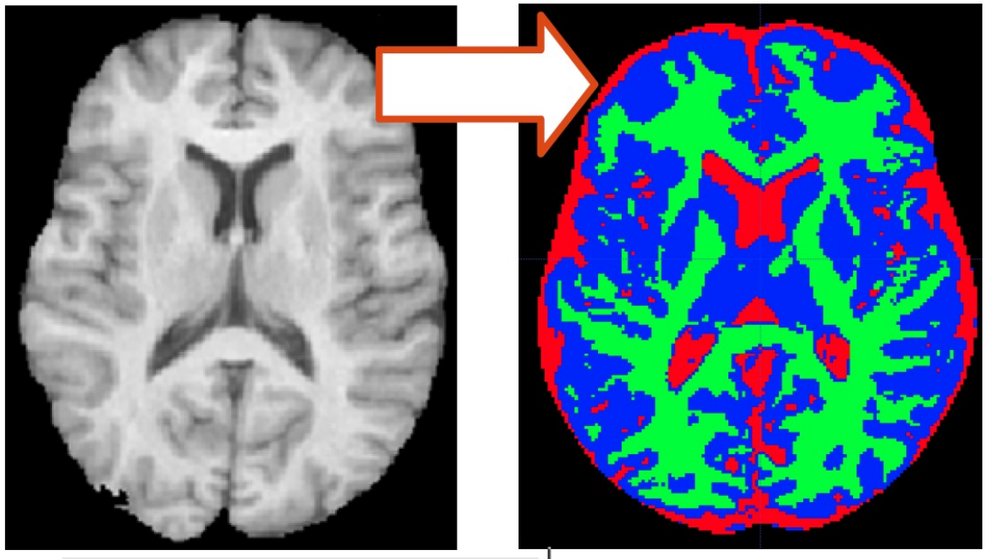
# Image Segmentation

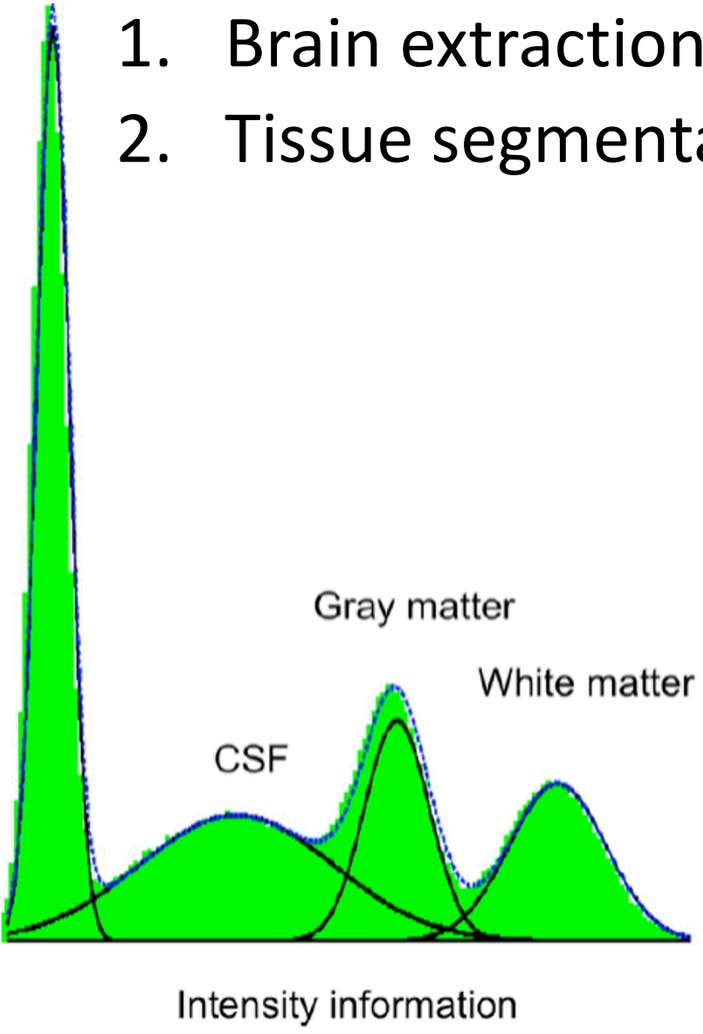
- Goal: locate all voxels for GM, WM, CSF
- Strategy (2 stages)
    1. Brain extraction from head MR image

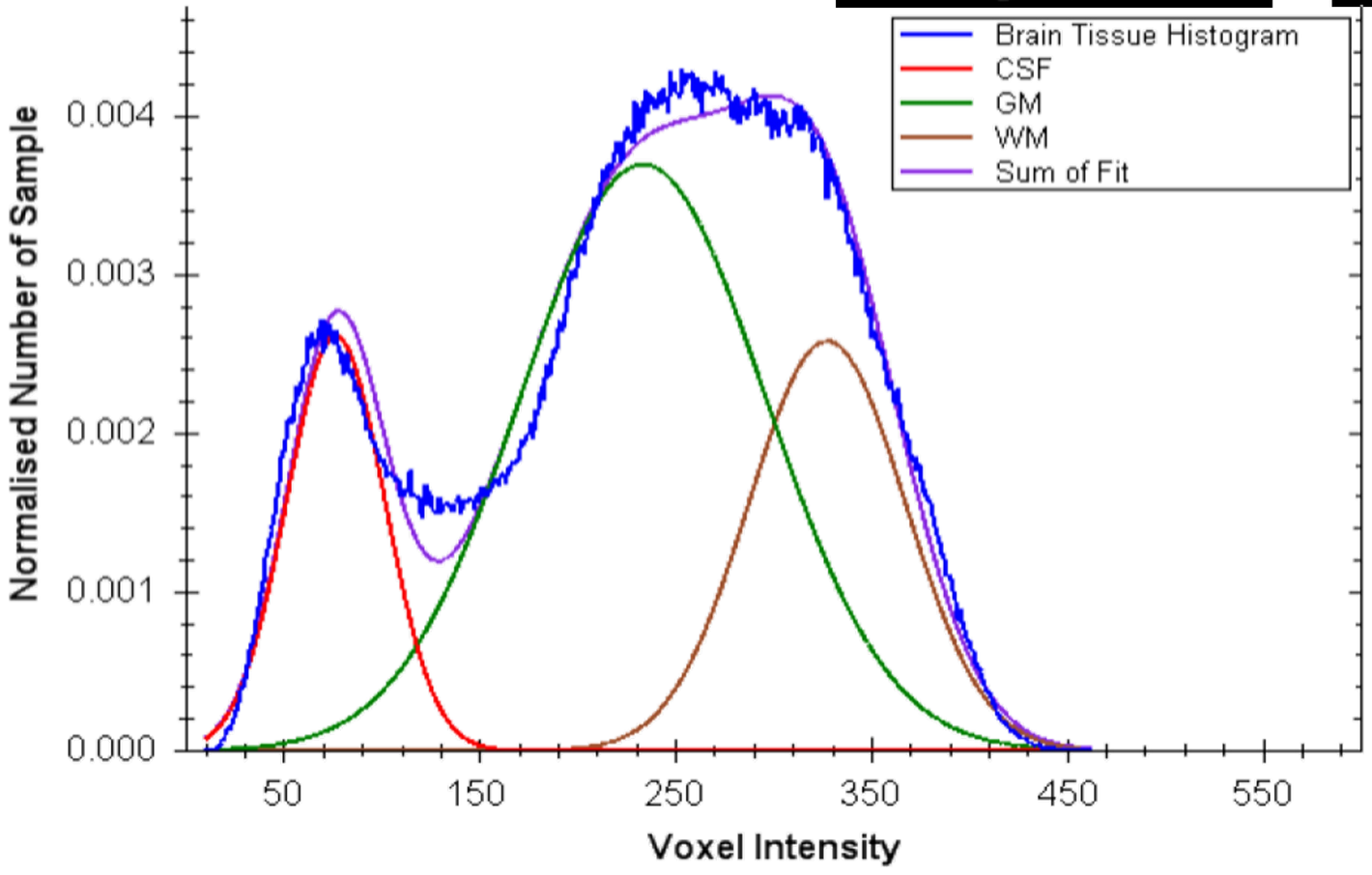


# Image Segmentation

- Goal: locate all voxels for GM, WM, CSF
- Strategy (2 stages)
  1. Brain extraction from head MR image
  2. Tissue segmentation within brain

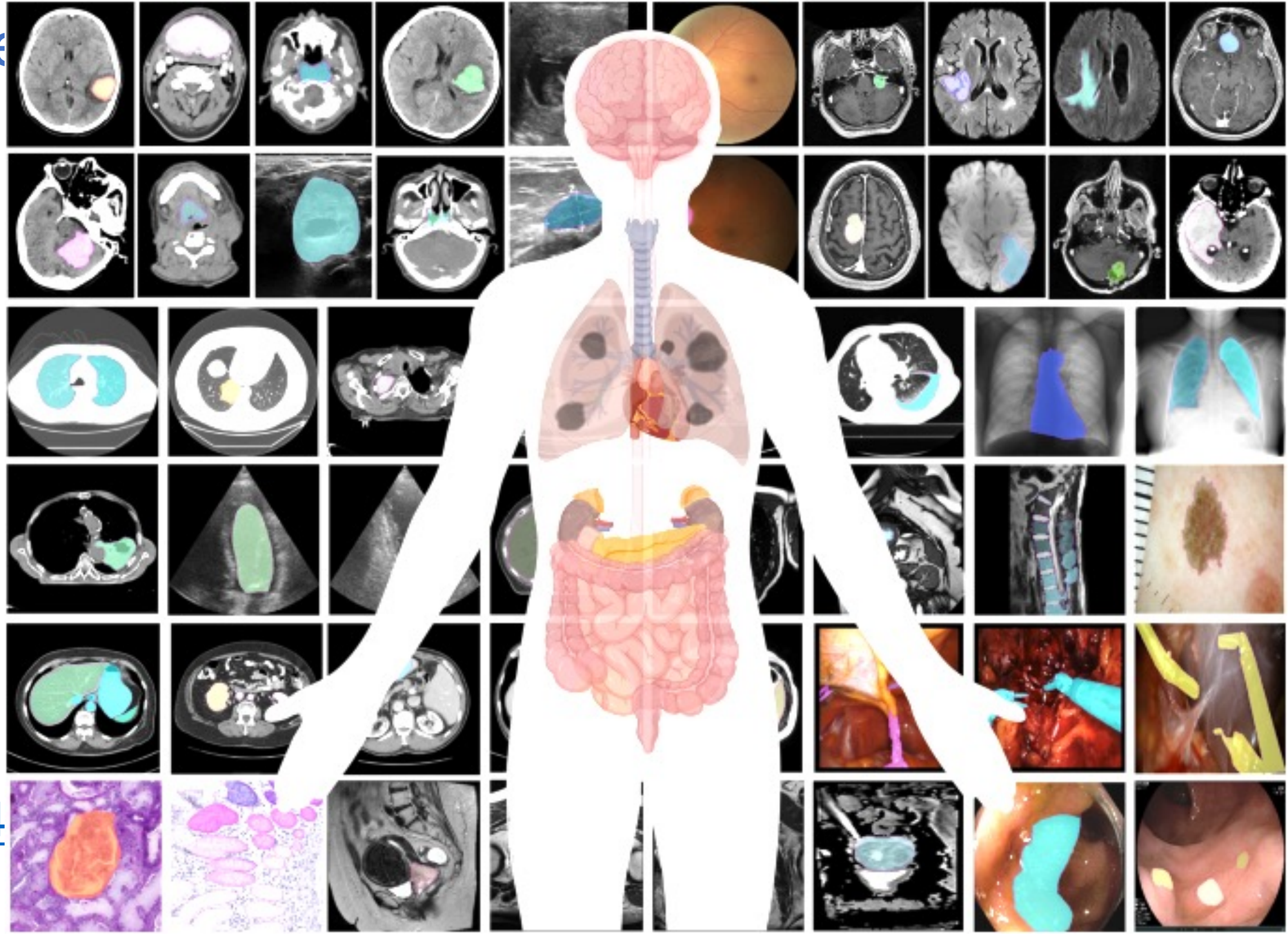




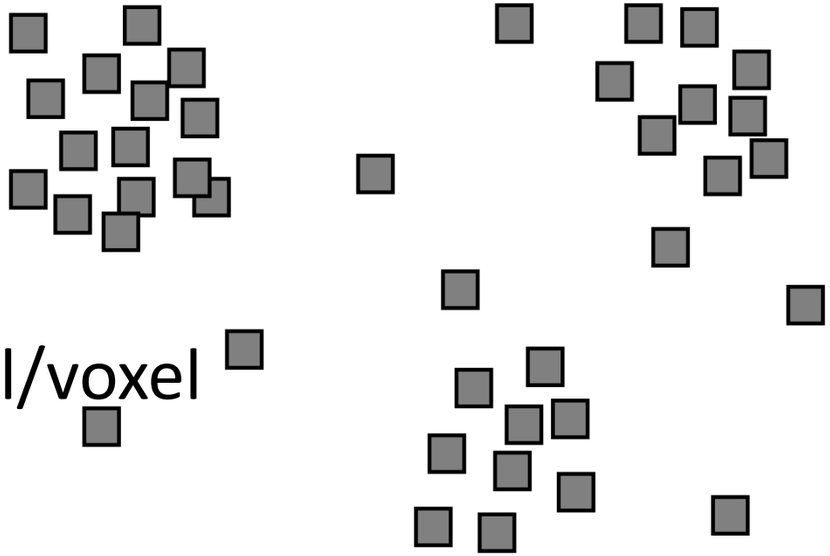
# Image Segme

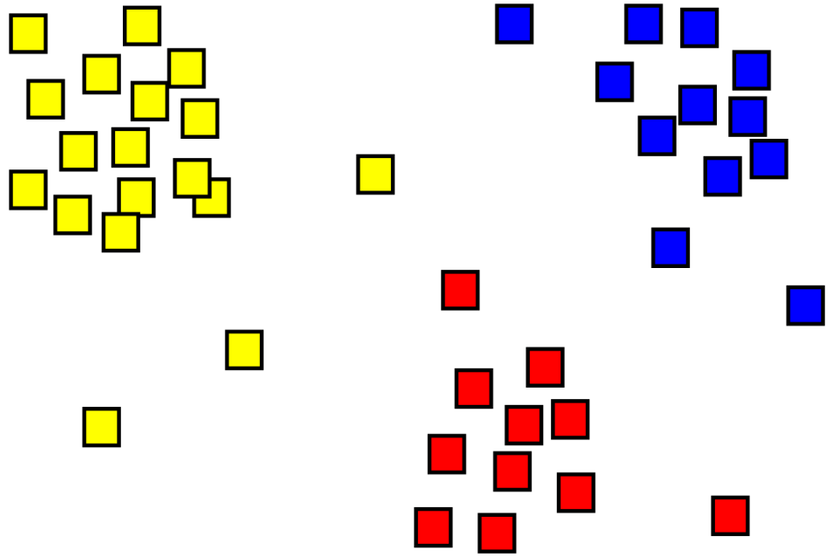
- “Segment anything in medical images”, Nature Communica-tions, 2024
- MedSAM
- www.nature.com/articles/s41467-024-44



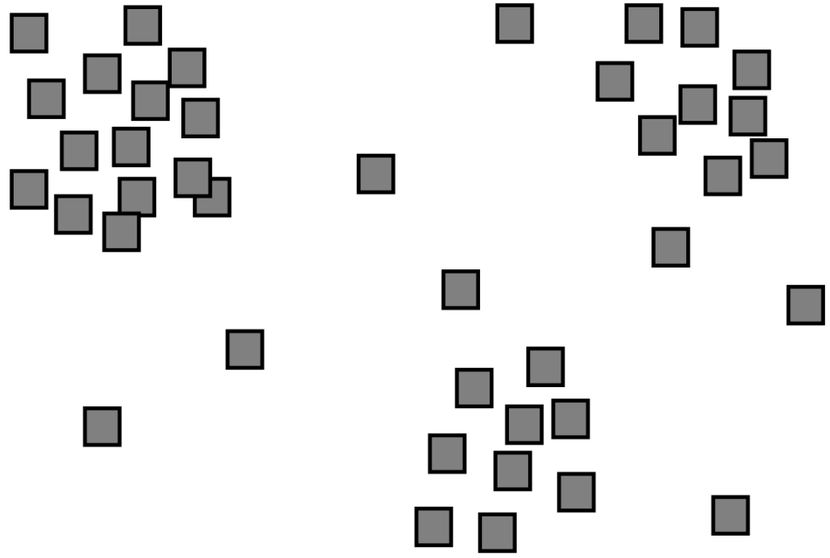
# K-means Clustering

- Given: data points in D-dimensional space
  - Data points are features, hand-crafted/learned, simple/complex, to model texture around each pixel/voxel
- Properties of data distribution
  - Data points lie within k clusters; are also referred to as “classes”/“groups”
    - Value of k is known/fixed
  - Data points belonging to the same cluster are more “similar” to each other, compared to data points belonging to different clusters
- Goal: partition data points into k clusters
  - Label each point to belong to one (and exactly one) of the clusters
    - Hard (discrete) decision made for each data point





# K-means Clustering



- Consider k clusters/groups $S_1$, ..., $S_k$
- Consider a representative point for each cluster
  - For i-th cluster, let the representative be $\mu_i$
  - Doesn't need to be one of the data points
- Optimization problem
  - Find: optimal clusters and their representatives
  - Penalize sum of squared distances of each data point to the representative of the cluster it is assigned to
  - Optimize representative $\mu_i$ and labels $S_i$ by minimizing:

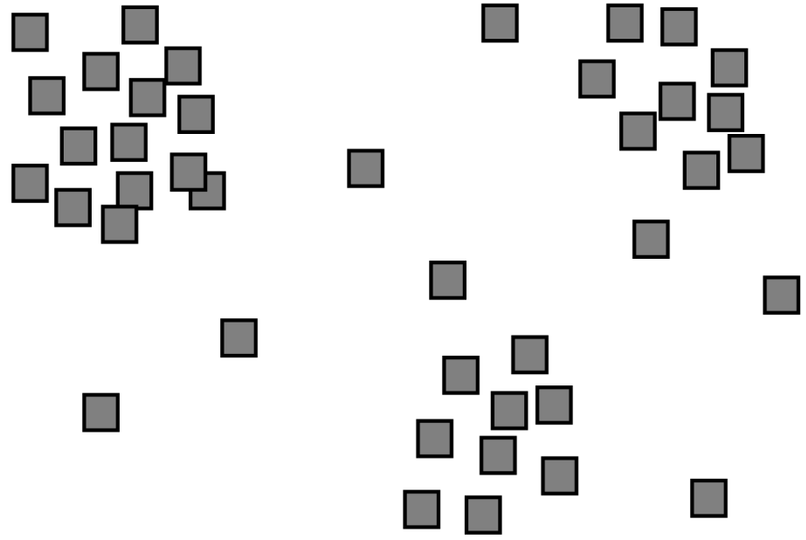
$$\sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

# K-means Clustering

$$\sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

- Optimization algorithm (2-step algorithm)
  1. An initial estimate for the set of representatives is given
     - Can equivalently assume an initial estimate for the clusters
  2. Given representatives $\mu_i$ , what are the optimal clusters $S_i$ ?
     - Which cluster will you assign to each data point to aim to decrease the objective function ?
     - For a data point, assign that cluster whose representative is closest to the data point
  3. Given clusters $S_i$ , what are the optimal representatives $\mu_i$ ?
     - For a specific cluster, what representative will minimize the penalty for that cluster ?
     - Assign representative = average of the data points in that cluster
     - Hence, representative is called the “mean”
  4. If objective function doesn’t decrease from previous estimate to the updated estimate, then terminate.
     Otherwise, go back to step 2.

# K-means Clustering

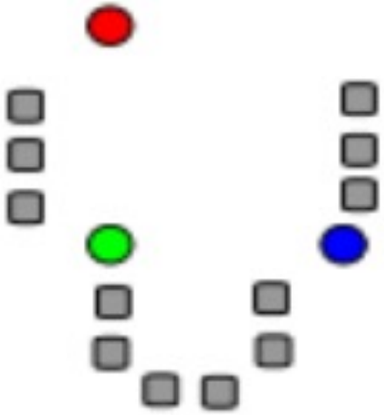


- Optimization algorithm
  - Is it guaranteed to converge/terminate ?
    - Yes.
      Data sample size is finite.
      So, number of possible labelings is finite.
      An infinite loop would imply revisiting the same labelling,
      but with different values of objective function (which is impossible).
  - Will it converge to a global optimum ? Why or why not ?
    - No guarantees.
    - See a counter example soon
  - Is the optimization problem convex ?
    - No. Combinatorial optimization. NP hard.
    - Fixing clusters (labelings),
      objective function for means is quadratic

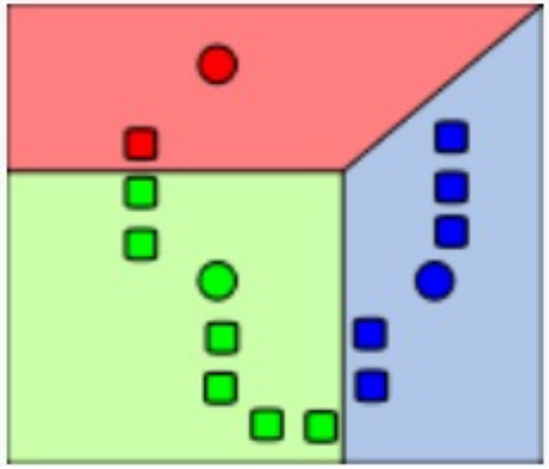
$$\sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

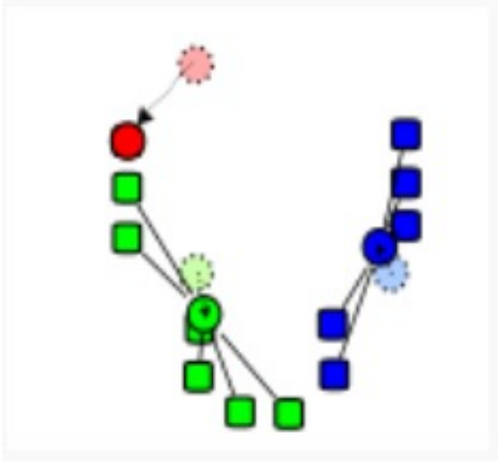
# K-means Clustering

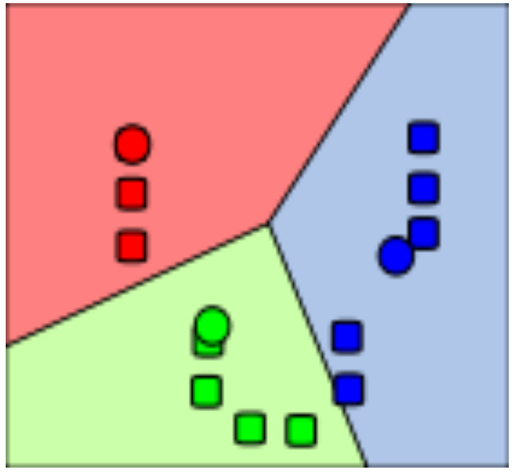
- K-means example
  - Data points shown in gray. K = 3 (fixed).
  - Initial means shown in R, G, B
    - From list of data points: take first K, or randomly select some K
    - Generate random K points in the space
    - Can we do better that any of these ? Yes. Will see soon.



- Given means, update clusters



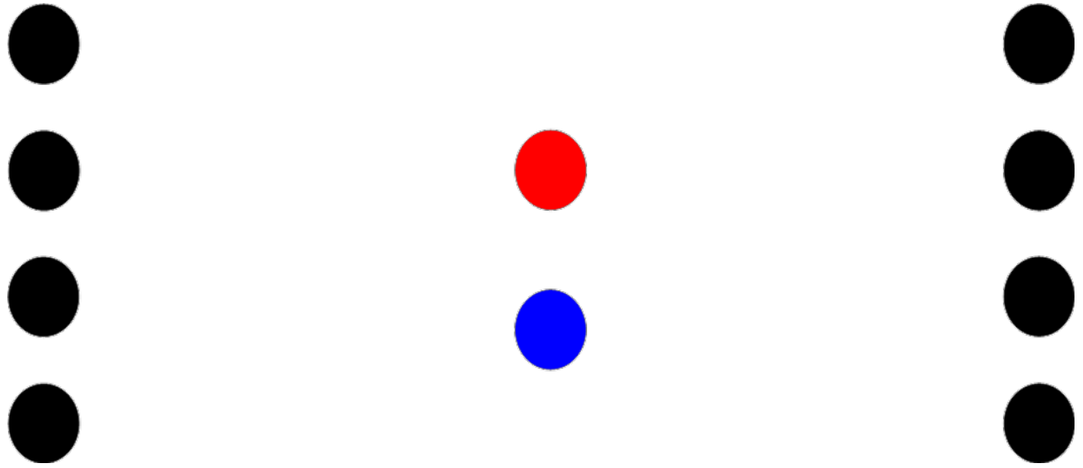




- Given clusters, update means

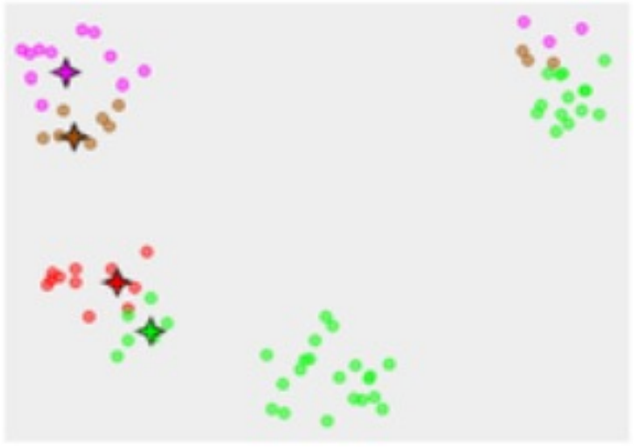
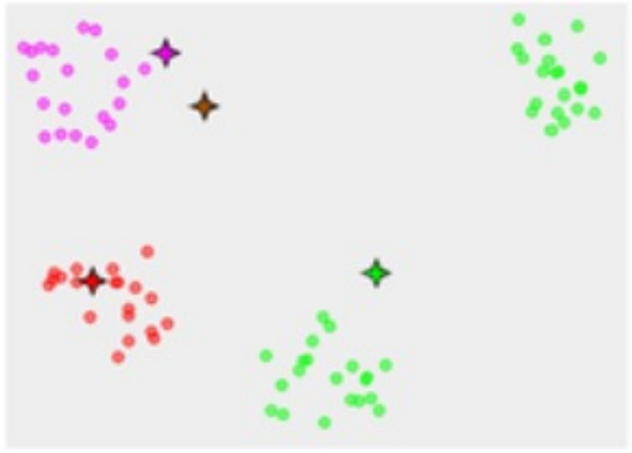
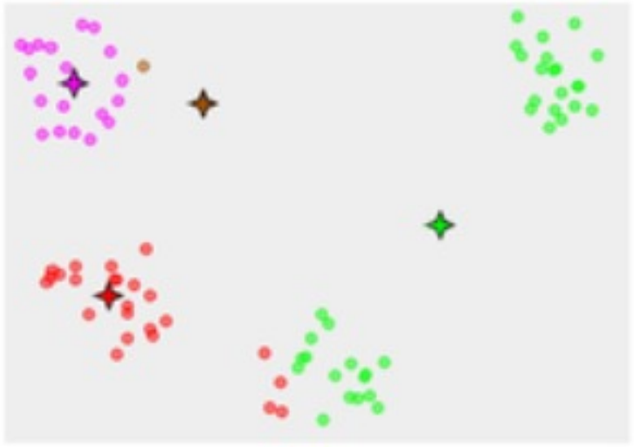
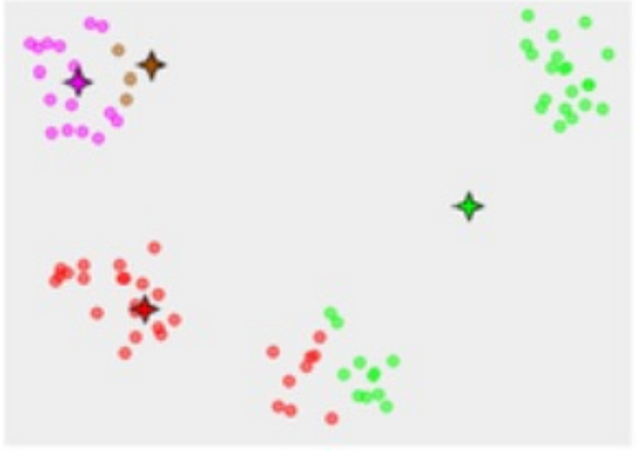
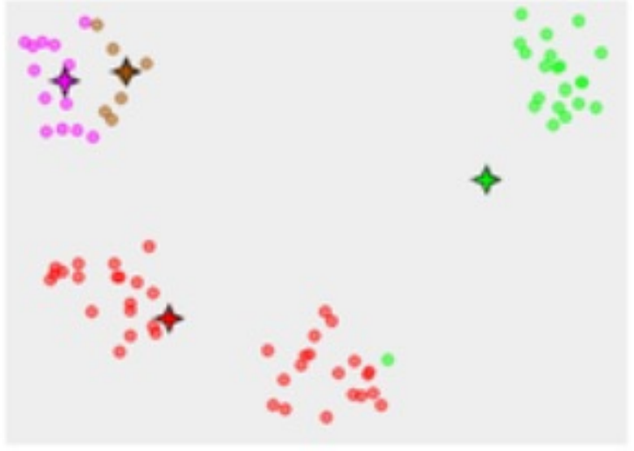
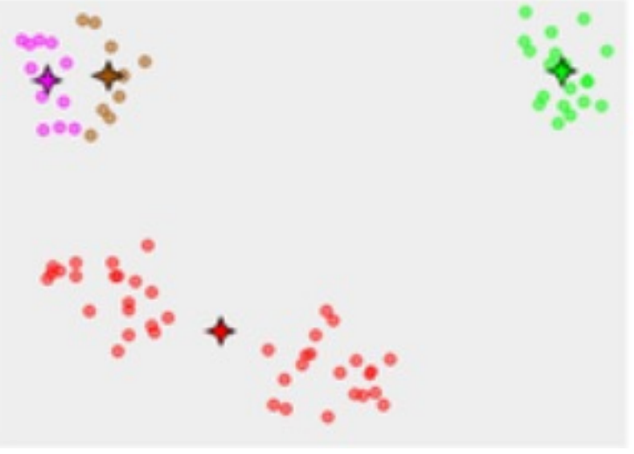
# K-means Clustering

- Convergence to local minimum
  - Example 1
  - K = 2



# K-means Clustering

- Convergence to local minimum
  - Example 2. K = 4.
  - Each picture shows 1 pass through data
    - First update means, then update labels

# K-means Clustering

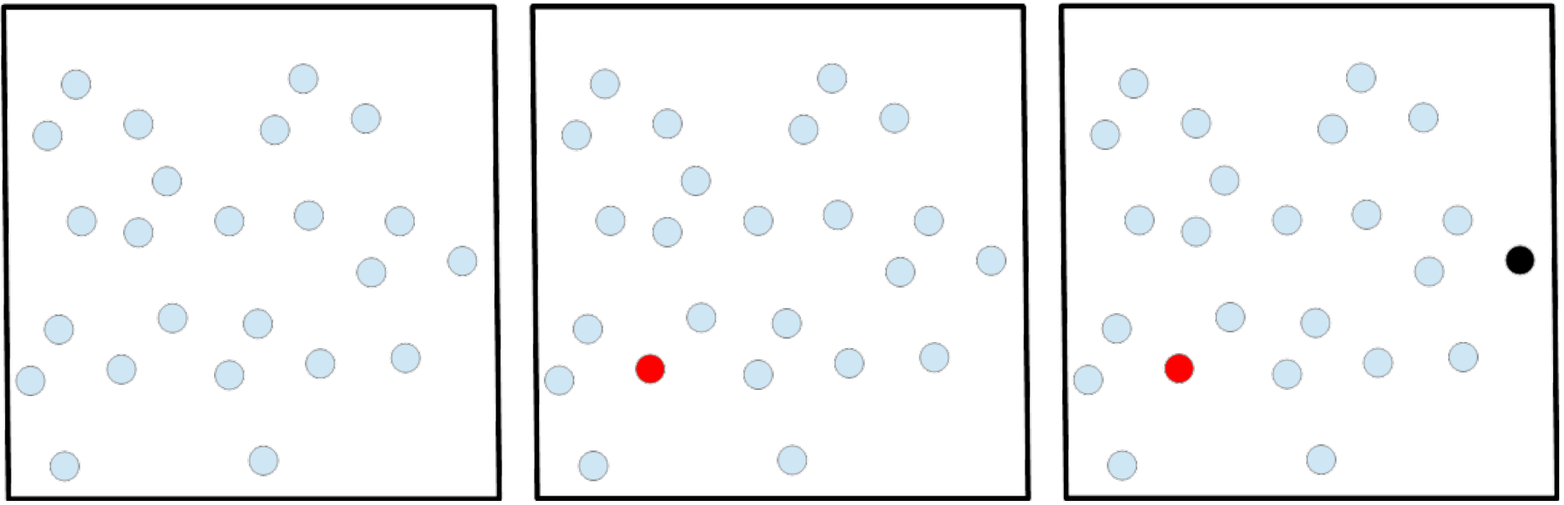
- Hope to get global optimum ?
  - Brute-force approach
    - Check objective-function value for all possible clusterings/labelings
    - N data points and K clusters → $K^N$ possible clusterings/labelings
  - Re-run algorithm using different initial mean estimates
    - Introduce randomization in the initialization
  - Better initialization
    - Will see some strategies

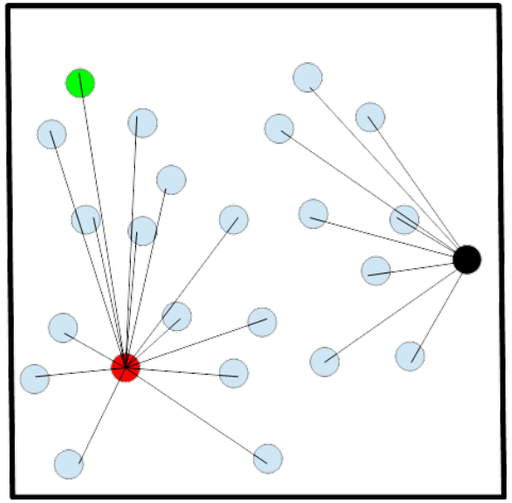
# K-means Clustering

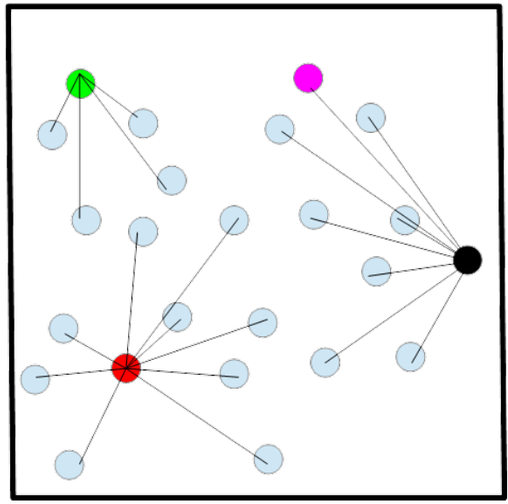
- Better initialization
  - Motivation: we want initial means to be spread across the data distribution to increase chances of initializing one representative in each cluster
  - Farthest-point clustering
  - Algorithm
    - **Given**: data points, value of ‘K'
    - Select first “mean” at random from among the data points
    - Select the other means, among the data points, iteratively, as follows
    - Assume: at some iteration, (n-1) means have been selected
    - Select the **n-th mean** to be the **data point** that maximizes the minimum distance among distances to the current (n-1) means

# K-means Clustering

- Better initialization
  - Farthest-point clustering
  - Select the point that is farthest from the current means







# K-means Clustering

- Better initialization
  - Farthest-point clustering: sensitive to outliers
  - K-means++
    - Don’t pick farthest point
    - Pick point ‘x’ with probability directly related to minimum distance

pick $x \in S$ uniformly at random and set $T \leftarrow \{x\}$
while $|T| < k$:
  pick $x \in S$ at random, with probability proportional to $\text{cost}(x, T) = \min_{z \in T} \|x - z\|^2$
  $T \leftarrow T \cup \{x\}$

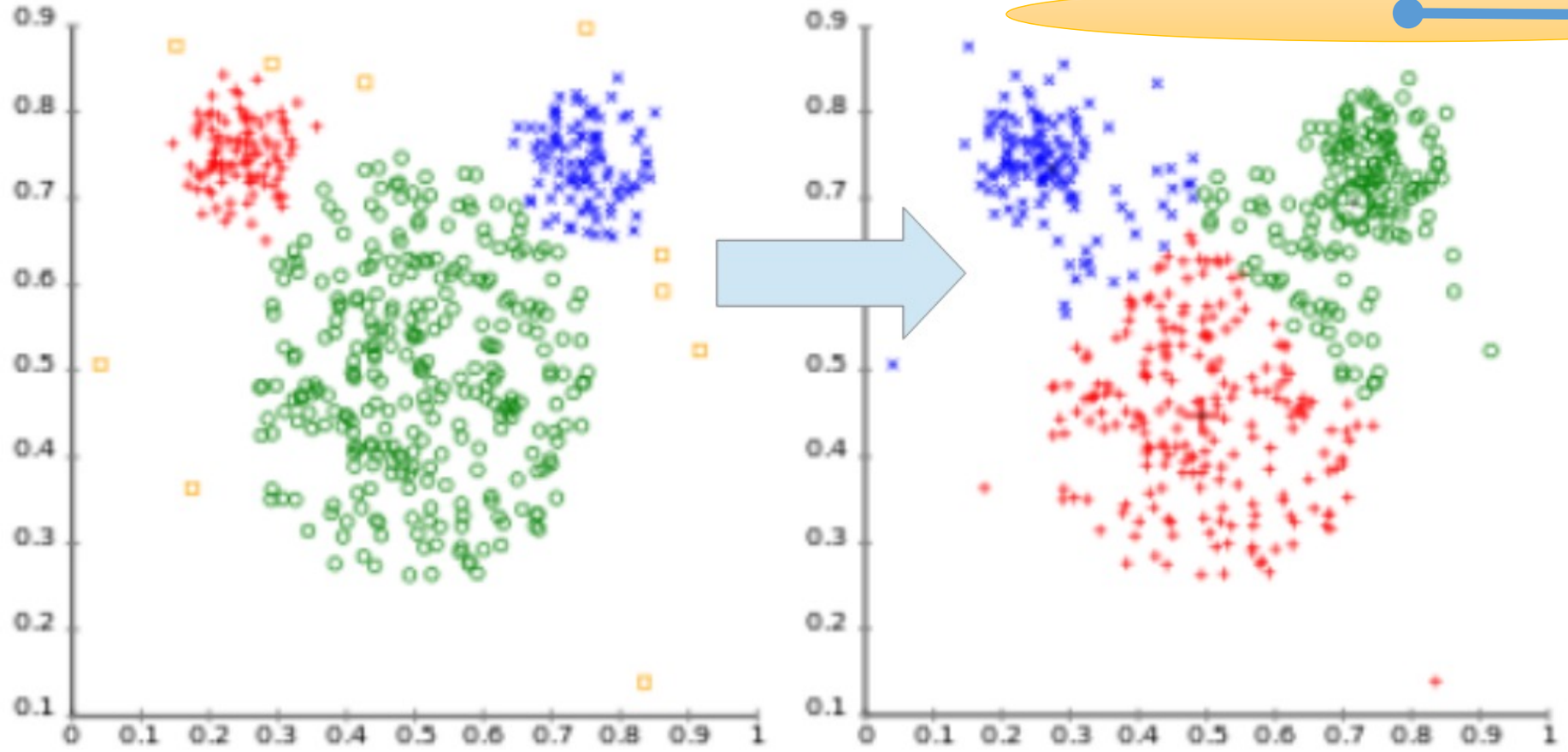
- k-means++: the advantages of careful seeding.
  Arthur D and Vassilvitskii S.
  In ACM-SIAM Symposium on Discrete Algorithms (SODA) 2007.

**Theorem 4.** *Let $T$ be the initial centers chosen by k-means++. Let $T^*$ be the optimal centers. Then $\mathbb{E}[\text{cost}(T)] \leq \text{cost}(T^*) \cdot O(\log k)$, where the expectation is over the randomness in the initialization procedure.*

- https://cseweb.ucsd.edu/~dasgupta/291-geom/kmeans.pdf

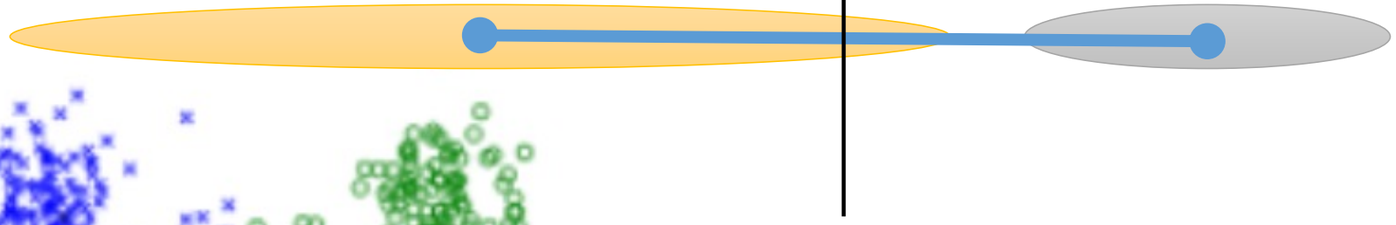
# K-means Clustering

- **Limitations**
  1. Crisp / hard assignments of each data point to cluster
  2. Tendency to produce clusters with equal spreads
     - Doesn't model (adapt to) variance/spread of the cluster



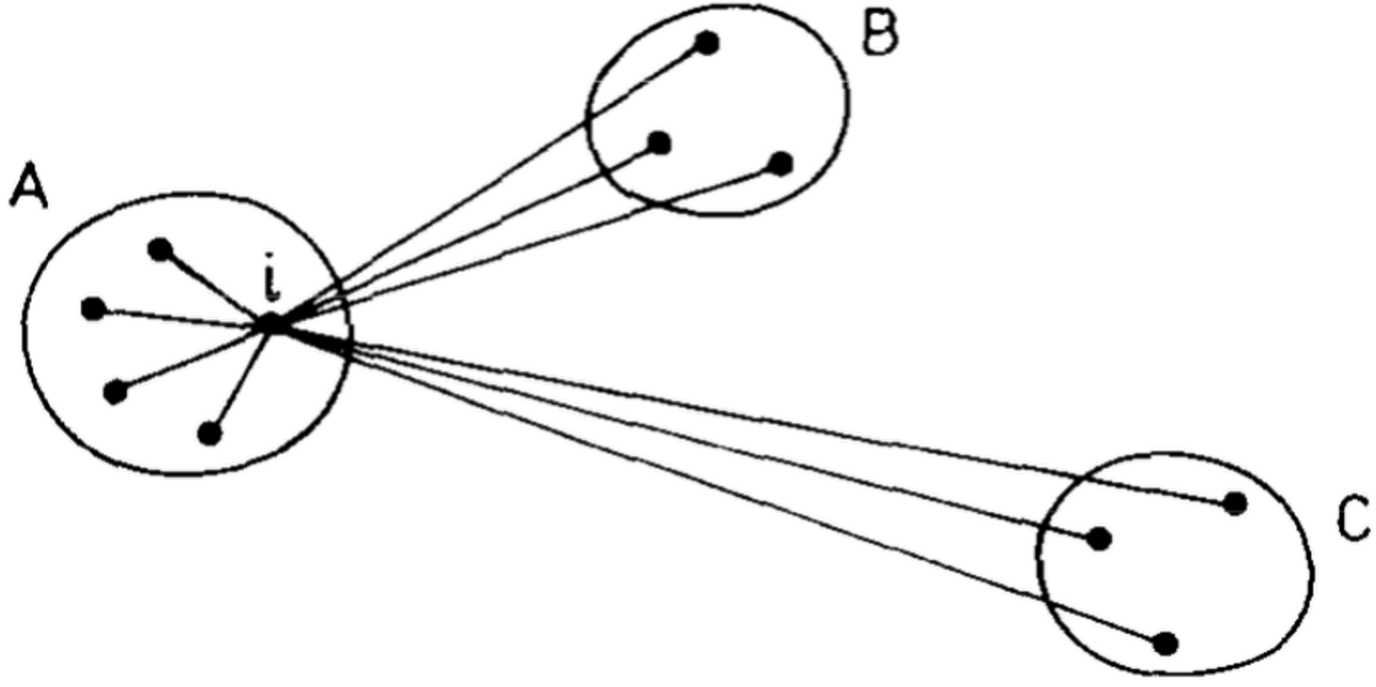






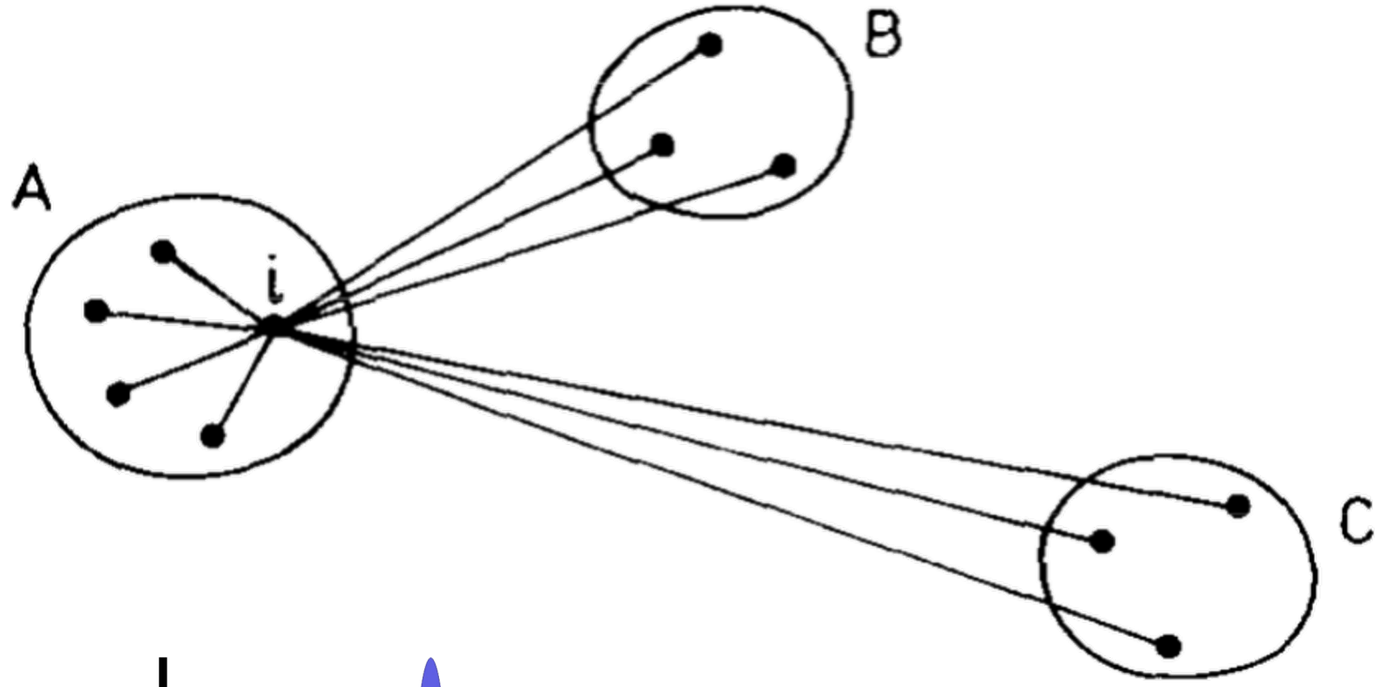
# Evaluating Quality of Clustering

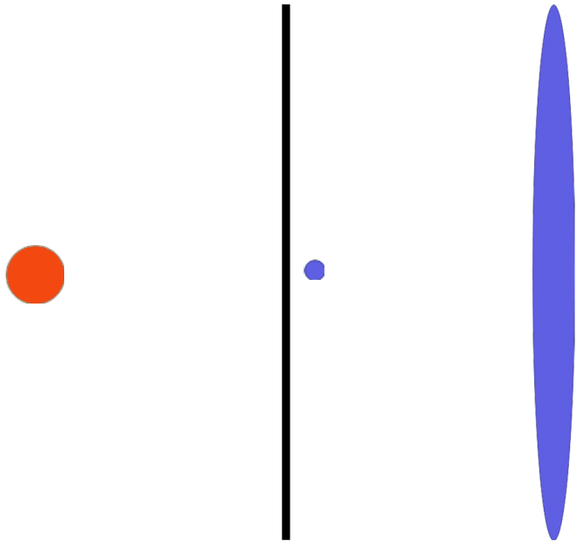
- "Silhouette" analysis [ J. Computational and Applied Math. 1987 ]
  - http://dx.doi.org/10.1016%2F0377-0427%2887%2990125-7
  - Measures how contained is each datum within its cluster
  - Leads to a visualization of the clustering
- For each datum $x_i$ assigned to cluster A
  - $a_i$ = average of distances between $x_i$ & $x_j$ within cluster A
    - $a_i > 0$
    - Small $a_i$ → more compact cluster
  - $b_i$ = minimum, over all clusters B, of average distance between $x_i$ & $x_j$ within another cluster B
    - $b_i > 0$
    - Large $b_i$ → more compact cluster



# Evaluating Quality of Clustering

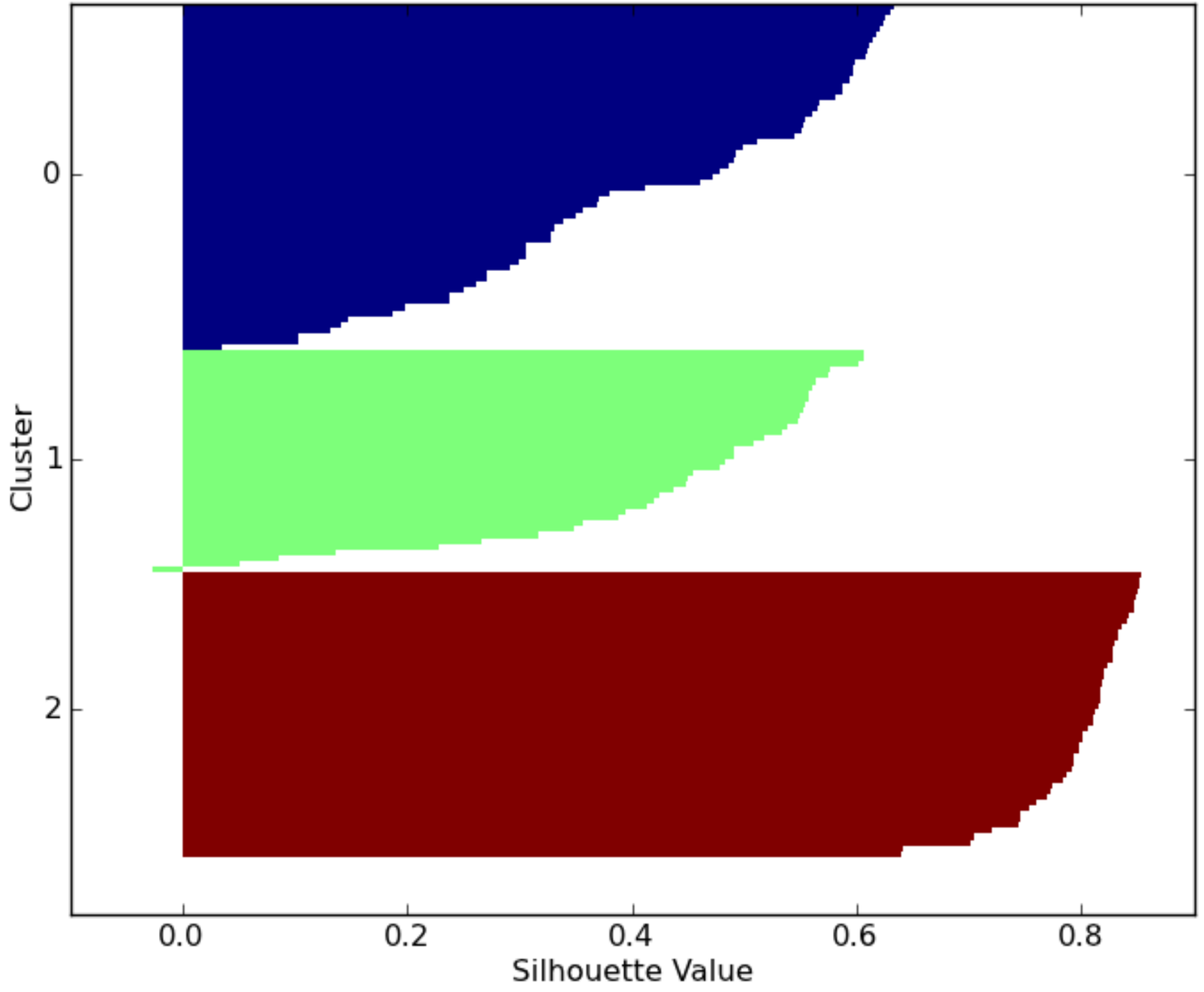
- For each datum, define $s_i := (b_i - a_i) / \max(a_i, b_i)$
  - When $b_i > a_i$
    - $s_i = 1 - a_i / b_i$
    - $0 < s_i <= 1$
  - When $b_i = a_i$
    - $s_i = 0$
  - When $b_i < a_i$
    - $s_i = b_i / a_i - 1$
    - $-1 <= s_i < 0$
  - We desire $s_i$ close to +1
    - i.e., $a_i << b_i$
  - When can $s_i$ be negative ?
    - A k-means based example



# Evaluating Quality of Clustering

- For each cluster, visualize graph of $s_i$ over i
  - This is the “silhouette” / outline
  - May be used to tune free parameters, e.g., number of clusters K



# Fuzzy-C-Means (FCM) Clustering

- Generalizes K-means clustering
  - Assigns, to each data point, a membership to belong to each cluster
    - Membership within range [0,1]
    - For each data point, sum of memberships over all clusters = 1
  - Produces a “soft”/“fuzzy” segmentation
    - K-means produces a “hard”/“crisp” segmentation

# Fuzzy-C-Means (FCM) Clustering

- Given
  - Data = { $y_j$ }, j = 1,...,N
  - Number of clusters = K (known / fixed)
- Memberships
  - $u_{jk}$ = membership (non-negative) of j-th point in k-th cluster
  - For each datum, over all classes k, memberships $u_{jk}$ sum to 1
- Objective function to be minimized
  - Penalize distance of datum j from mean of class k
  - Weight penalty based on membership $u_{jk}$

$$\sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^{q}(y_j - c_k)^2$$

- q > 1: free parameter controlling fuzziness of clusters/memberships
- Constraints
  - Positivity constraint on memberships gets satisfied automatically

$$\forall j, \sum_k u_{jk} = 1$$

# Fuzzy-C-Means (FCM) Clustering

$$\sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^{q}(y_j - c_k)^2$$

$$\forall j, \sum_k u_{jk} = 1$$

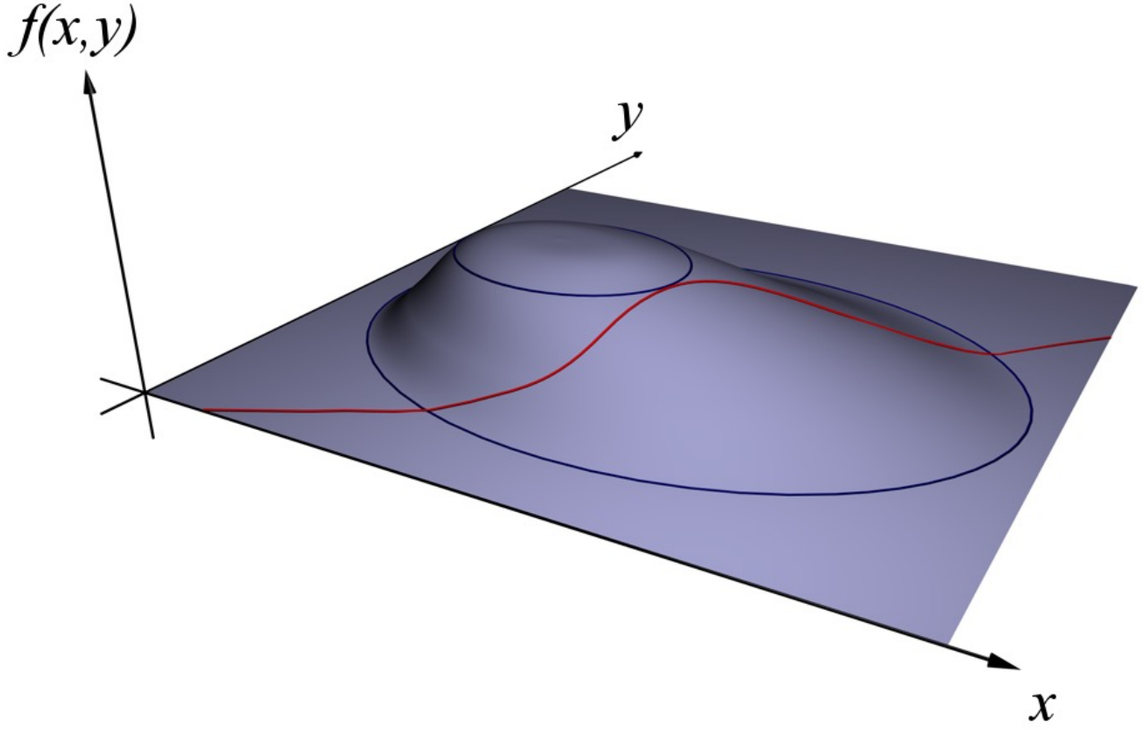
- FCM reduces to K-means
  - What happens if you force $u_{jk}$ to be crisp ?
    - $u_{jk}$ = 1 for exactly one cluster k
    - $u_{jk'}$ = 0 for all other clusters k'
    - What happens to the objective function ?
      - Objective function becomes exactly the same as that for K means

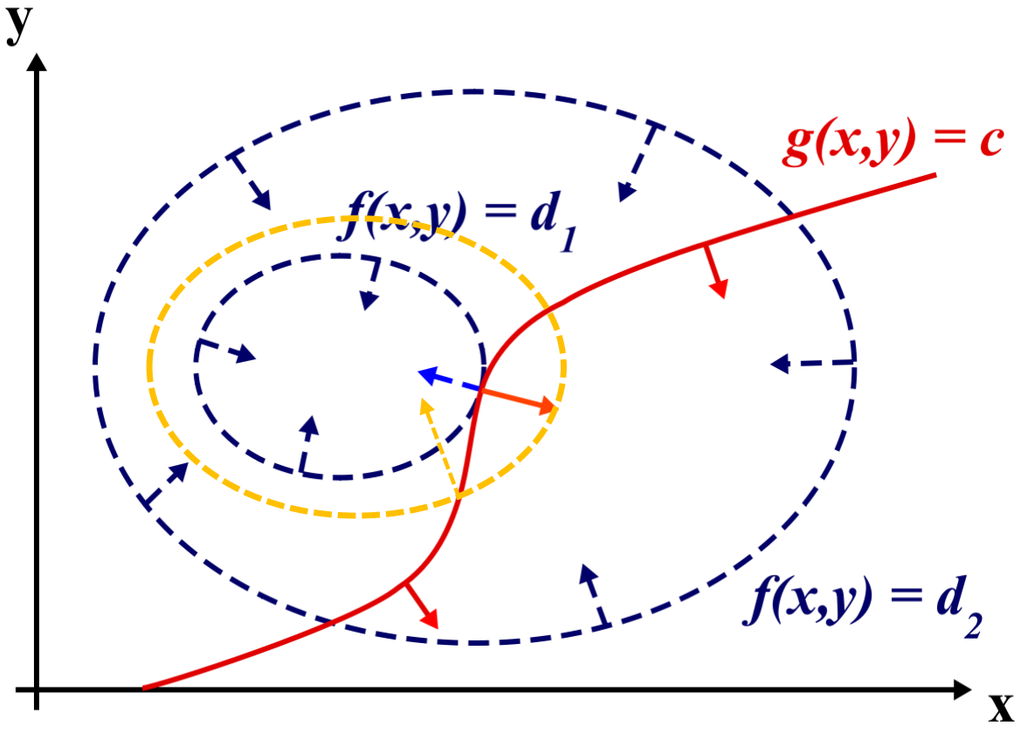
# Fuzzy-C-Means (FCM) Clustering

$$\sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^{q}(y_j - c_k)^2$$

$$\forall j, \sum_k u_{jk} = 1$$

- Constrained nonlinear optimization
  - Objective function type: nonlinear
  - Constraint type: equality
  - Approach: method of Lagrange multipliers
    - Minimize f(x,y) subject to equality constraint g(x,y)=c
    - What should happen at the optimal solution (x*,y*) ?



# Fuzzy-C-Means (FCM) Clustering

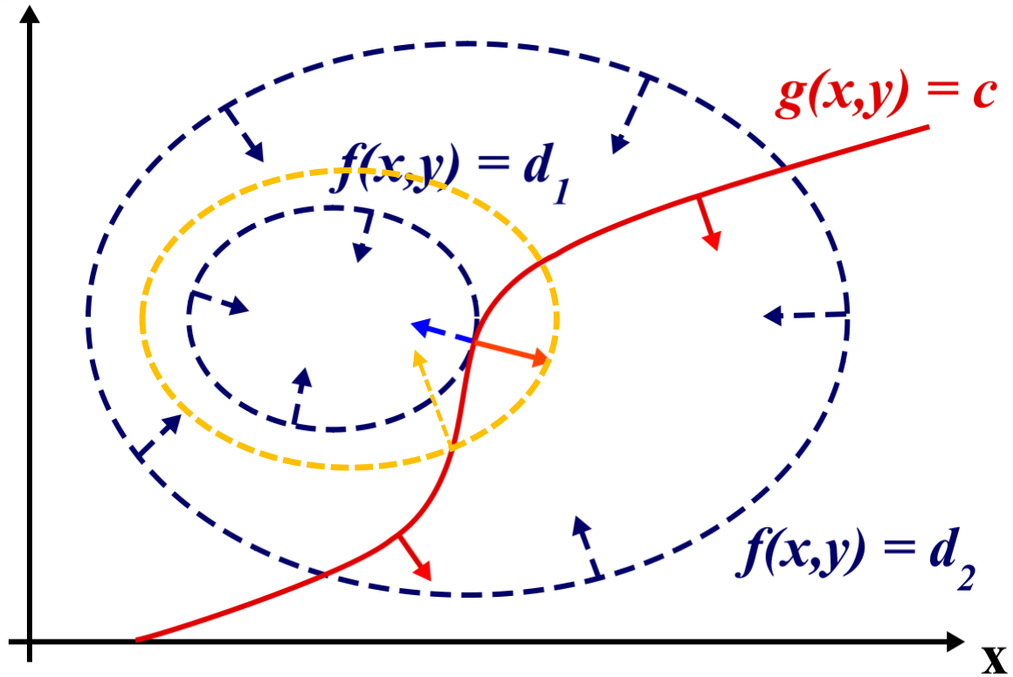
- Method of Lagrange multipliers
  - What should happen at the optimal solution (x*,y*) ?
  1. Constraint must be satisfied: g(x*,y*) = c
  2. At (x*,y*),
     gradient of objective function f(.)
     must be parallel to
     gradient of constraint-related function g(.)

$$\nabla_{x,y} f = -\lambda \nabla_{x,y} g,$$

for some $\lambda$

where

$$\nabla_{x,y} f = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right), \qquad \nabla_{x,y} g = \left( \frac{\partial g}{\partial x}, \frac{\partial g}{\partial y} \right)$$



# Fuzzy-C-Means (FCM) Clustering

- Method of Lagrange multipliers
  - Introduces a function called the Lagrangian
    - $L(x,y,\lambda) = f(x,y) + \lambda ( g(x,y) - c )$
    - $\lambda$ is called the Lagrange parameter
  - Then, at the optimum, the following conditions should hold for x, y, λ
    - Partial derivatives of Lagrangian w.r.t. variables (x, y, λ) should be zero
    - Solve for 3 variables using 3 resulting equations
    - Partial derivative w.r.t. 'x' and 'y' gives us back the “parallel” condition
      - $df(x,y) / dx = - \lambda\, dg(x,y) / dx$
      - $df(x,y) / dy = - \lambda\, dg(x,y) / dy$
    - Partial derivative w.r.t. 'λ' gives us back the constraint
      - $g(x,y) = c$

# Fuzzy-C-Means (FCM) Clustering

- Joseph-Louis Lagrange
  - Advisors: Euler + …
  - Students: Fourier, Poisson, …
  - It was not until he was 17
    that he showed any taste for math
  - For 20 years in Berlin,
    except for a short time when he was ill,
    he produced (on average) 1 paper per month
  - Lagrange is one of the
    72 prominent French scientists
    who were commemorated on plaques
    at the first stage of Eiffel Tower

# Fuzzy-C-Means (FCM

- Joseph-Louis Lagrange

## Lagrange's [edit]

- Lagrange's approximation theorem
- Lagrange's formula (disambiguation)
- Lagrange's identity (disambiguation)
- Lagrange's theorem (group theory)
- Lagrange's theorem (number theory)
- Lagrange's four-square theorem
- Lagrange's trigonometric identities

## Lagrangian [edit]

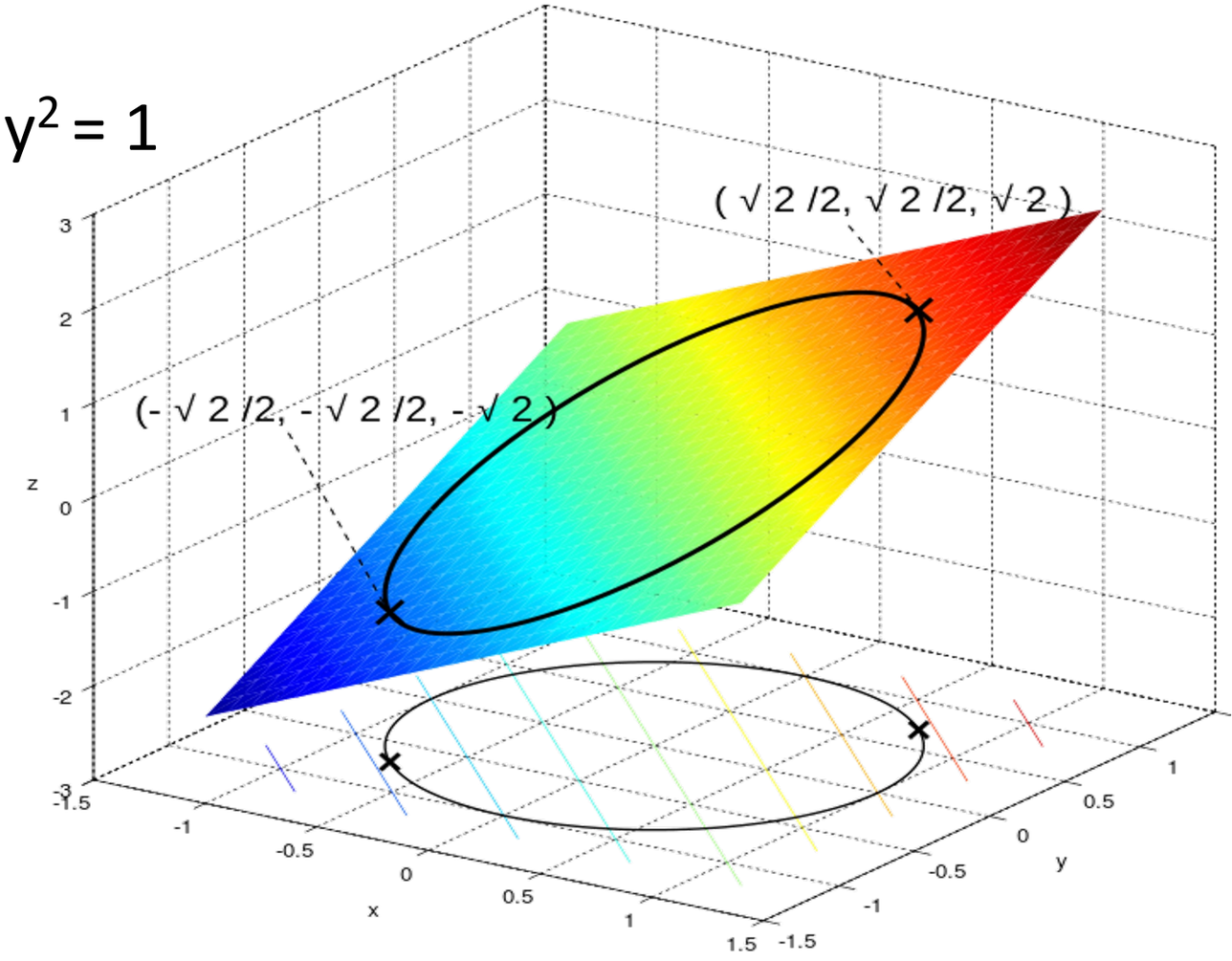
- Lagrangian
- Lagrangian analysis
- Lagrangian coordinates
- Lagrangian derivative
- Lagrangian drifter
- Lagrangian foliation
- Lagrangian Grassmannian
- Lagrangian intersection Floer homology
- Lagrangian mechanics
- Lagrangian mixing
- Lagrangian point
- Lagrangian relaxation
- Lagrangian submanifold
- Lagrangian subspace
- Nonlocal Lagrangian
- Proca lagrangian
- Special Lagrangian submanifold

## Lagrange [edit]

- Euler–Lagrange equation
- Green–Lagrange strain
- Lagrange bracket
- Lagrange–d'Alembert principle
- Lagrange error bound
- Lagrange form
- Lagrange interpolation
- Lagrange invariant
- Lagrange inversion theorem
- Lagrange multiplier
- Lagrange number
- Lagrange point colonization
- Lagrange polynomial
- Lagrange property
- Lagrange reversion theorem
- Lagrange resolvent
- Lagrange stream function

# Fuzzy-C-Means (FCM) Clustering

- Method of Lagrange multipliers
  - Lagrangian optimization gives a necessary condition
  - Not a sufficient condition
  - Example:
    maximize $f(x,y) = x + y$ subject to $x^2 + y^2 = 1$
  - How does constraint
    look like (in 2D XY plane) ?
  - How do objective-function contours
    look like (in 2D XY plane) ?



# Fuzzy-C-Means (FCM) Clustering

$$\sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^{q}(y_j - c_k)^2$$

- FCM optimization
  1. Start with an initial estimate for memberships
  2. Fixing memberships, solve for cluster means
     - Quadratic function minimization. Closed-form solutic

$$\forall j, \sum_k u_{jk} = 1$$

  3. Fixing cluster means, solve for memberships
     - Use method of Lagrange multipliers
     - Lagrangian

$$L(\{u_{jk}\},\{c_k\},\{\lambda_j\}) := \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^{q}(y_j - c_k)^2 + \lambda_j\left(\sum_{k=1}^{K} u_{jk} - 1\right)$$

       - We don't really need L(.) to be a function of cluster means, but it doesn't harm us either
     - Denote
       - $d_{jk} := (y_j - c_k)^2$
       - Squared distance of j-th data point from k-th cluster mean
  4. Go back to step 2

# Fuzzy-C-Means (FCM) Clustering

- FCM optimization
  - Fixing cluster-means (or $d_{jk}$), solve for memberships and Lagrange multipliers

$$\frac{\partial L(\{u_{jk}\},\{\lambda_j\},\{c_k\})}{\partial u_{jk}} = 0 = q u_{jk}^{q-1}(y_j - c_k)^2 + \lambda_j$$

$$u_{jk} = \left(\frac{-\lambda_j}{q d_{jk}}\right)^{\frac{1}{q-1}}$$

$$\frac{\partial L(\{u_{jk}\},\{\lambda_j\},\{c_k\})}{\partial \lambda_j} = 0 = \sum_k u_{jk} - 1$$

$$\sum_k \left(\frac{-\lambda_j}{q d_{jk}}\right)^{\frac{1}{q-1}} = 1 \implies (-\lambda_j)^{\frac{1}{q-1}} = \frac{1}{\sum_k \left(\frac{1}{q d_{jk}}\right)^{\frac{1}{q-1}}}$$

$$u_{jk} = \frac{\left(\frac{1}{d_{jk}}\right)^{\frac{1}{q-1}}}{\sum_k \left(\frac{1}{d_{jk}}\right)^{\frac{1}{q-1}}}$$

$$L(\{u_{jk}\},\{c_k\},\{\lambda_j\}) := \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^q (y_j - c_k)^2 + \lambda_j \left(\sum_{k=1}^{K} u_{jk} - 1\right)$$

$$d_{jk} := (y_j - c_k)^2$$

# Fuzzy-C-Means (FCM) Clustering

- FCM optimization
  - Update for memberships
    - Positivity constraint met
    - Sum-to-1 constraint met

$$u_{jk} = \frac{\left(\frac{1}{d_{jk}}\right)^{\frac{1}{q-1}}}{\sum_k \left(\frac{1}{d_{jk}}\right)^{\frac{1}{q-1}}}$$

  - Interpretation
    - If data point j has a larger distance $d_{jk}$ to cluster mean $c_k$, then its membership $u_{jk}$ to cluster k reduces
    - If the user-defined parameter $q \to \infty$, then $\forall j, \forall k$ memberships $u_{jk}$ are constant, i.e., $u_{jk} = 1 / K$
    - If the user-defined parameter $q \to 1$, then membership $u_{jk} = 1$ iff $d_{jk} < d_{jk'}$ , otherwise $u_{jk} = 0$
      - Hard / crisp clustering (equivalent to K means)

$$L(\{u_{jk}\}, \{c_k\}, \{\lambda_j\}) := \sum_{j=1}^{N} \sum_{k=1}^{K} u_{jk}^q (y_j - c_k)^2 + \lambda_j \left( \sum_{k=1}^{K} u_{jk} - 1 \right) \qquad d_{jk} := (y_j - c_k)^2$$

# Fuzzy-C-Means (FCM) Clustering

- FCM optimization
  - Fixing memberships, solve for class means

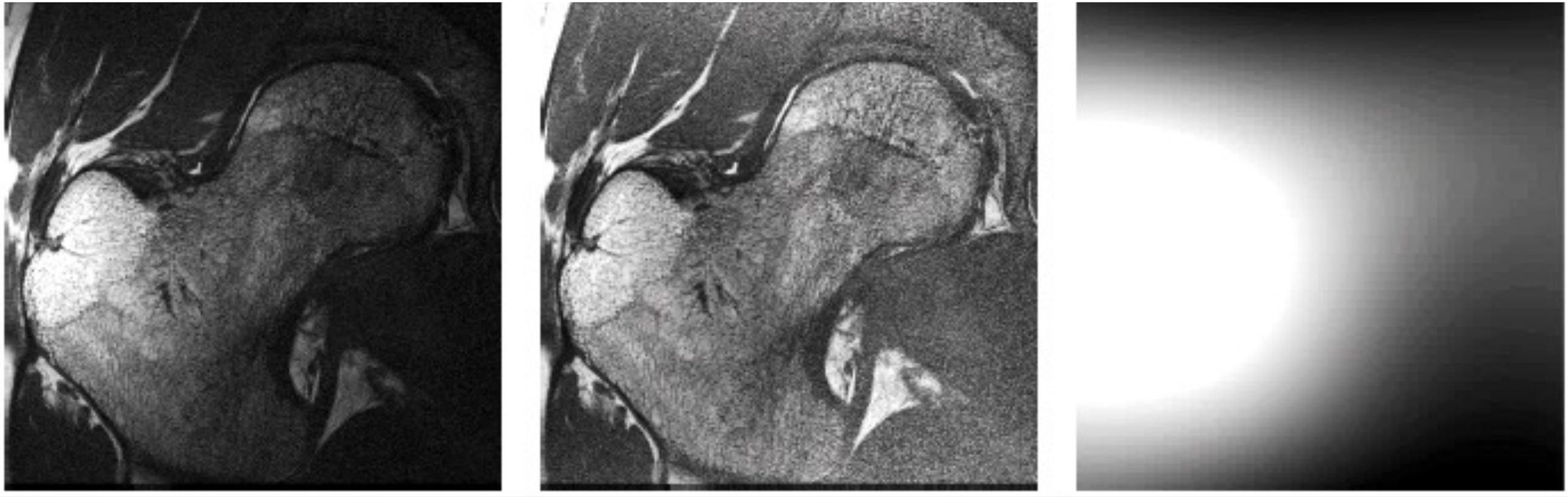
$$\frac{\partial L(\{u_{jk}\},\{\lambda_j\},\{c_k\})}{\partial c_k} = 0 = \sum_{j=1}^{N} u_{jk}^q 2(y_j - c_k) \longrightarrow c_k = \frac{\sum_{j=1}^{N} u_{jk}^q y_j}{\sum_{j=1}^{N} u_{jk}^q}$$

  - Interpretation
    - Class mean is weighted average of data points
    - Weight larger ← membership larger
  - What happens with memberships are crisp ?

$$L(\{u_{jk}\},\{c_k\},\{\lambda_j\}) := \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^q (y_j - c_k)^2 + \lambda_j \left(\sum_{k=1}^{K} u_{jk} - 1\right) \qquad d_{jk} := (y_j - c_k)^2$$

# Tissue Segmentation in Brain MRI

- Inhomogeneity field ("bias" field) in MRI images



**Left:** A slice from hip MRI data acquired with a surface coil. **Center:** Same slice after coil correction. **Right:** The estimated bias field.

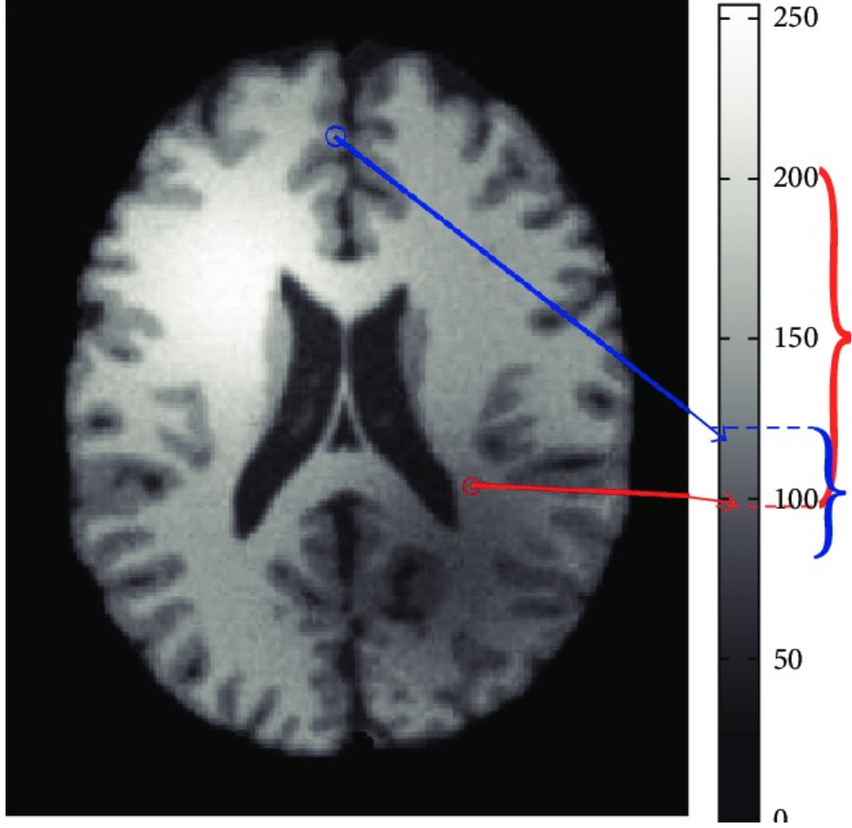
https://radiology.ucsf.edu/research/labs/musculoskeletal-quantitative-imaging/research-directions/folkesson-coil-correction

- Spatial distribution of b1 magnetic field is imperfect within slice
- Spatially-smooth multiplicative corruption on intensities (magnetization)

# Tissue Segmentation in Brain MRI

- Vignetting
  - Reduction of brightness or saturation at periphery compared to image center
    - Often unintended/undesired ← camera settings / lens limits
    - Sometimes introduced to draw attention to frame center

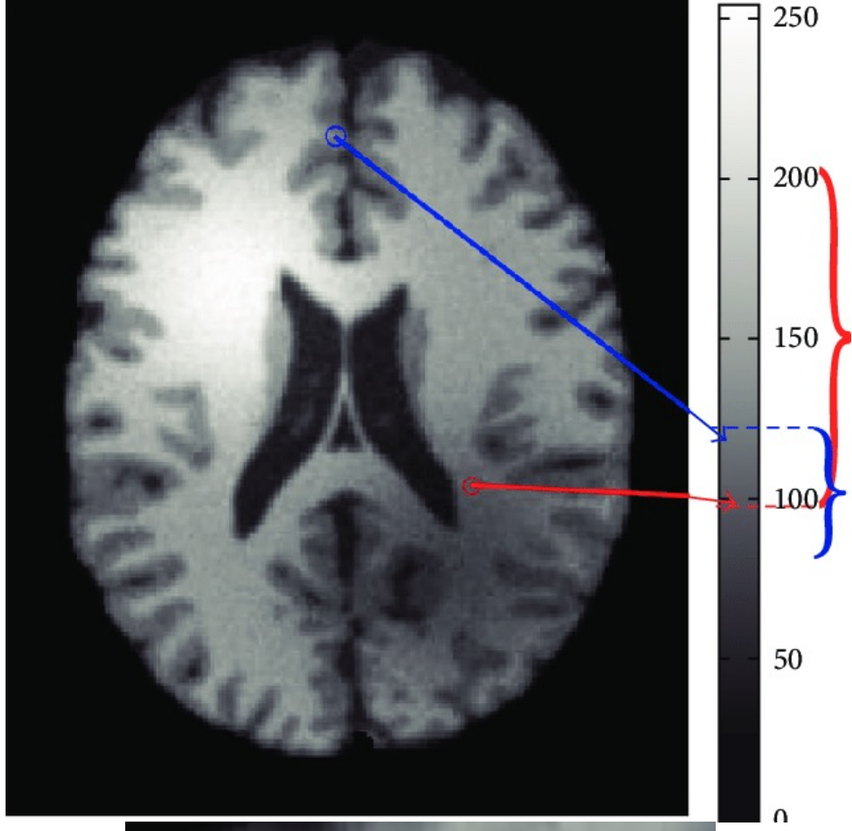

# FCM Seg. + Bias-Field Correction

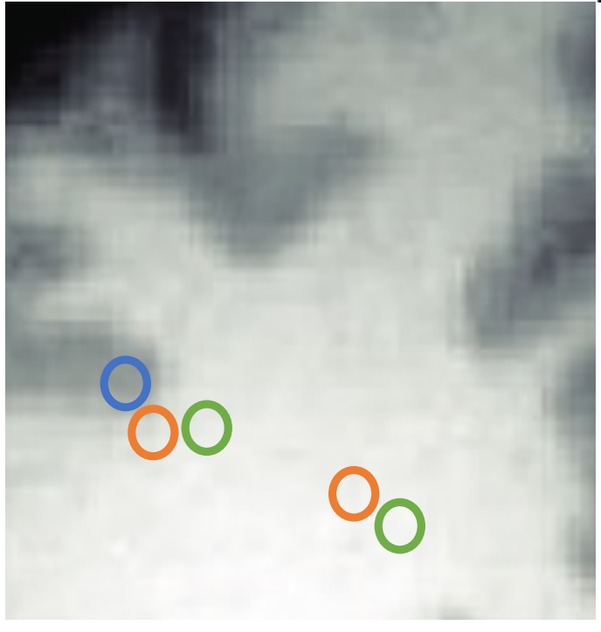



- MRI image (magnitude) with N voxels
- At voxel i, intensity is $x_i$ (unknown)
- At voxel i, bias-field is $b_i$ (unknown)
  - But varies smoothly over entire spatial domain (not just foreground/brain region), without any edges
- Noise model: i.i.d. additive zero-mean Gaussian
  - Approximation valid for voxels with high-SNR
  - Otherwise, can work with complex values, and reformulate problem
- At voxel i, observed intensity is $y_i := x_i\, b_i + \eta_i$, where noise $\eta_i \sim G(0,\sigma^2)$
- Bias field increases variance of intensities within each tissue

# FCM Seg. + Bias-Field Correction

- Two more model assumptions $y_i := x_i b_i + \eta_i$
  - Image comprises K classes (K is known)
  - Each class k has constant MRI (uncorrupted) intensity = $c_k$
  - Biologically reasonable
- Model assumptions lead to:
  - At voxel i,
    if its neighbor j belongs to class k,
    then, $x_j = c_k$ (piecewise constant tissue intensity)
    $b_j \approx b_i$ (spatial smoothness of bias)
    and, so, $x_j b_j \approx c_k b_i$



# FCM Seg. + Bias-Field Correction

- Strategy

$y_i := x_i b_i + \eta_i$

  - In neighborhood of voxel i: at voxel j, penalize difference between
    (i) observed neighborhood intensities $\mathbf{y_j}$ and
    (ii) intensities predicted by model $\mathbf{x_j b_j}$
  - If neighborhood voxel j belongs to class k, then, predicted intensity = $x_j b_j \approx \mathbf{c_k b_i}$
  - Because we don't know the class that voxel j is in,
    penalize differences for all classes weighted by membership $u_{jk}^q$ in that class
  - Bias-constancy is local, so weight penalty based on distance between voxels i,j
    - Weight $w_{ij} \rightarrow 0$ when distance > threshold. Weights sum to 1. Can choose as Gaussian.
      - Implement as neighborhood "mask". Choose width of Gaussian based on smoothness of bias field.
  - Objective function $J := \sum_{i=1}^{N} J_i$, where, per voxel i, $J_i := \sum_{j=1}^{N} w_{ij} \sum_{k=1}^{K} \left( u_{jk}^q (y_j - c_k b_i)^2 \right)$
  - Constraints on memberships: $\forall j; \sum_{k=1}^{K} u_{jk} := 1$

# FCM Seg. + Bias-Field Correction

$$J_i := \sum_{j=1}^{N} w_{ij} \sum_{k=1}^{K} \left( u_{jk}^q (y_j - c_k b_i)^2 \right)$$

- Objective function
  - Rewrite:

$$J := \sum_{i=1}^{N} J_i := \sum_{j=1}^{N} \sum_{k=1}^{K} u_{jk}^q \left( \sum_{i=1}^{N} w_{ij} (y_j - c_k b_i)^2 \right) \quad \text{constraints} : \forall j, \sum_k u_{jk} = 1$$

    - At voxel j, consider predictions based on biases $b_i$ at all neighboring voxels i
  - For FCM, recall:

$$\sum_{j=1}^{N} \sum_{k=1}^{K} u_{jk}^q (y_j - c_k)^2$$

- Denote distance between $y_j$ and mean $c_k$ as: $d_{kj} := \sum_{i=1}^{N} w_{ij} (y_j - c_k b_i)^2$
  - $d_{kj}$ depends only on class k & voxel-number j
- Rewrite optimization problem as:

$$\min_{\{u_{jk}\},\{c_k\},\{b_i\}} J(\{u_{jk}\},\{c_k\},\{b_i\}) := \sum_{j=1}^{N} \sum_{k=1}^{K} u_{jk}^q d_{kj}$$

$$\text{constraints} : \forall j, \sum_k u_{jk} = 1$$

# FCM Seg. + Bias-Field Correction

$$d_{kj} := \sum_{i=1}^{N} w_{ij}(y_j - c_k b_i)^2$$

- Optimization via Lagrange multipliers
  - Lagrangian

$$L(\{u_{jk}\},\{c_k\},\{b_i\},\{\lambda_j\}) := \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^q d_{kj} + \sum_{j=1}^{N} \lambda_j \left(1 - \sum_k u_{jk}\right)$$

  - Keeping class means and bias fixed, solve for **memberships** and **Lagrange multipliers**

$$\frac{\partial L(\{u_{jk}\},\{\lambda_j\},\{c_k\})}{\partial u_{jk}} = 0 = q u_{jk}^{q-1} d_{kj} - \lambda_i$$

$$\frac{dL(\{u_{jk}\},\{\lambda_j\},\{c_k\})}{d\lambda_i} = 0 = 1 - \sum_k u_{jk}$$

$$u_{jk} = \frac{\left(\frac{1}{d_{kj}}\right)^{\frac{1}{q-1}}}{\sum_k \left(\frac{1}{d_{kj}}\right)^{\frac{1}{q-1}}}$$

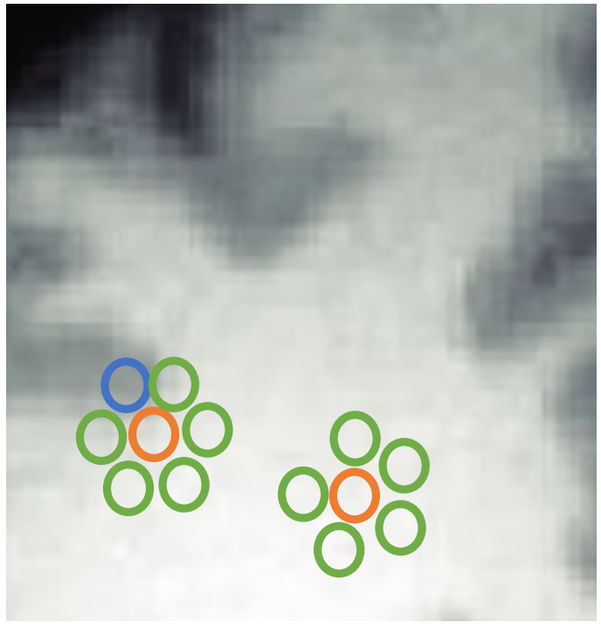
  - Compared to FCM, $d_{kj}$ generalizes squared distance over neighborhood
    - Involving averaging/smoothing across neighborhood
  - When there is NO bias or, equivalently, $b_i$ = constant, then formulation reduces to FCM

# FCM Seg. + Bias-Field Correction

- Solve for **bias**, keeping memberships, multipliers, and class means fixed

$$\frac{\partial L(\{u_{jk}\},\{c_k\},\{b_i\},\{\lambda_j\})}{\partial b_i} = 0 = \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^{q}\frac{\partial d_{kj}}{\partial b_i} = \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^{q} w_{ij} 2(y_j - c_k b_i) c_k$$

- Rearranging terms: $b_i = \dfrac{\sum_{j=1}^{N} w_{ij} y_j \sum_{k=1}^{K} u_{jk}^{q} c_k}{\sum_{j=1}^{N} w_{ij} \sum_{k=1}^{K} u_{jk}^{q} c_k^2}$
- Interpretation
  - If memberships are crisp then: $b_i \approx \sum_j w_{ij} y_j c_{k(j)} / \sum_j w_{ij} c_{k(j)}^2$
    - Assume bias field is smooth: $y_j = x_j b_j + n_j \approx c_{k(j)} b_i + n_j$
      - If data is noiseless, then all good (RHS reduces to $b_i$)
      - If data is noisy,
        then convolution with (Gaussian) mask w reduces effect of noise, keeping b smooth
      - If all neighbors are in class k, then bias $b_i$ = spatially-weighted average ratios of $y_j/c_k$



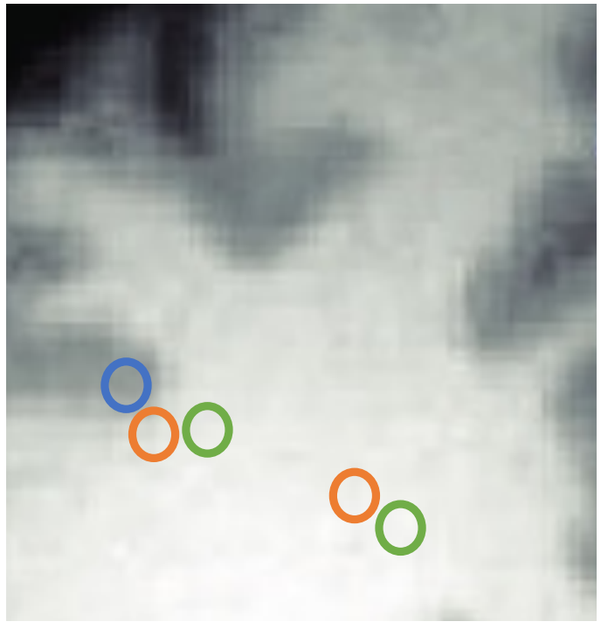
$$y_j := x_j\, b_j + \eta_j$$

$$L(\{u_{jk}\},\{c_k\},\{b_i\},\{\lambda_j\}) := \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^{q} d_{kj} + \sum_{j=1}^{N} \lambda_j \left(1 - \sum_k u_{jk}\right) \quad d_{kj} := \sum_{i=1}^{N} w_{ij}(y_j - c_k b_i)^2$$

# FCM Seg. + Bias-Field Correction

- Solve for **class means**, keeping memberships, multipliers, and bias fixed

$$\frac{\partial L(\{u_{jk}\},\{c_k\},\{b_i\},\{\lambda_j\})}{\partial c_k} = 0 = \sum_j u_{jk}^q \frac{\partial d_{kj}}{\partial c_k} = \sum_j u_{jk}^q \sum_i w_{ij} 2(y_j - c_k b_i)(-b_i)$$



- Rearranging terms: $c_k = \frac{\sum_j u_{jk}^q y_j \sum_i w_{ij} b_i}{\sum_j u_{jk}^q \sum_i w_{ij} b_i^2}$
- Interpretation
  - Bias field b gets convolved/smoothed with (Gaussian) mask w
  - If bias field is constant = f,
    then $c_k$ = membership-weighted mean of ratios $y_j/f$
  - If (memberships are close-to crisp) and
    (bias field varies slowly over neighborhoods),
    then: $c_k \approx \sum_{j \in C^k} y_j b_j / \sum_{j \in C^k} b_j^2$

$y_j := x_j\, b_j + \eta_j$

$$L(\{u_{jk}\},\{c_k\},\{b_i\},\{\lambda_j\}) := \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^q d_{kj} + \sum_{j=1}^{N} \lambda_j \left(1 - \sum_k u_{jk}\right) d_{kj} := \sum_{i=1}^{N} w_{ij}(y_j - c_k b_i)^2$$

# FCM Seg. + Bias-Field Correction

- Model: $y_i := x_i\, b_i + \eta_i$
  - If voxel i belongs (100% membership) to class k, then $y_i := c_k\, b_i + \eta_i$
  - Did we formulate the problem right ? Is the model identifiable ? i.e., do class means & bias field have unique solutions ?
  - What happens when, say, noise is absent ?

$$y_j := x_j\, b_j + \eta_j$$

$$L(\{u_{jk}\},\{c_k\},\{b_i\},\{\lambda_j\}) := \sum_{j=1}^{N}\sum_{k=1}^{K} u_{jk}^{q} d_{kj} + \sum_{j=1}^{N} \lambda_j \left(1 - \sum_{k} u_{jk}\right) \quad d_{kj} := \sum_{i=1}^{N} w_{ij}(y_j - c_k b_i)^2$$

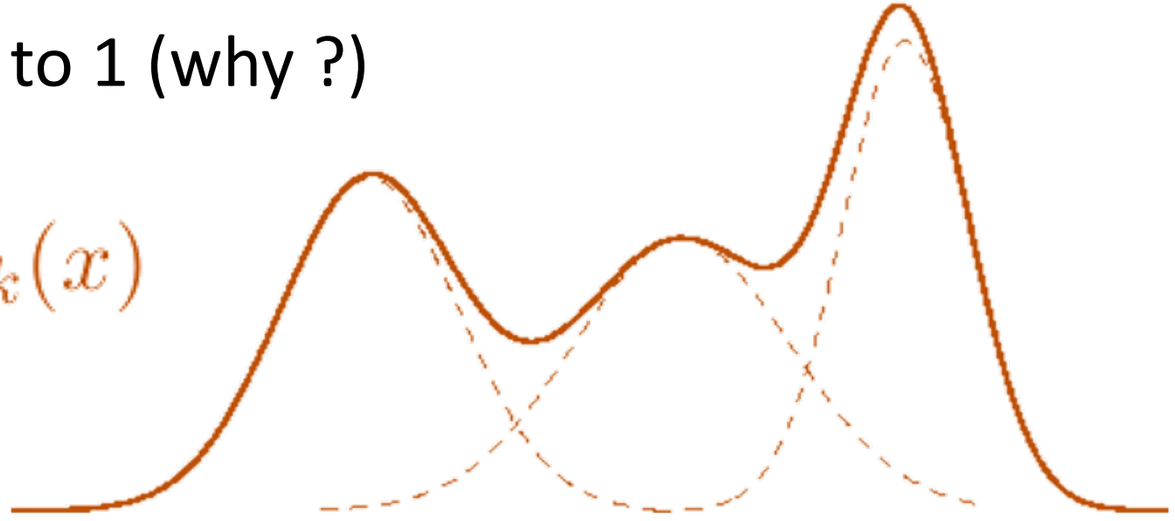
# Limitations of K-means, FCM

- Cannot model clusters with **unequal spreads**
  - How to do that ?
    Model variance of each cluster
- Cannot model **ellipsoidal** cluster distribution (in multi-D space)
  - How to do that ?
    Multivariate Gaussian model (covariance matrix) for each cluster
- Don't enforce **spatial smoothness** on segmentation
  - How to do that ?
    MRF model on membership image
- Cannot model **multimodal** clusters or '**curved**' distribution of a cluster
  - How to do that ?
    Nonlinear modelling using, e.g., kernels, deep learning

# Mixture Model

- What is a mixture model ?
  - Statistical model to represent a general multi-modal PDF
  - Assumes data drawn from multiple subgroups (mixture) within a population
- Models distribution as a convex combination of standard distributions
  - Weights $w_k$ are positive (why ?) and sum to 1 (why ?)

$$p(x) := \sum_{k=1}^{K} w_k p_k(x)$$



- How to interpret weights ?
  - Larger weight $w_k$ → data more likely to be derived from k-th component
- How to generate data from such a model ?
  - First draw a component k (with probability $w_k$); then draw x from $p_k(.)$

# Mixture Model

- When number of PDFs K is finite
  - Finite mixture model
- When each PDF is a Gaussian
  - Gaussian mixture model (GMM)
  - A GMM can model uni-modal / multi-modal / curved distributions

- If we fit a Gaussian to 2D data that lies close to a circle, then is that a good model fit ?

# Gaussian Mixture Model

- Fitting a GMM to data
  - **Given**: data y := { $y_n$ } where n=1,…,N
    - Each observation $y_n$ is drawn independently of the others
  - **Goal**: fit a GMM with K components
    - $P(x) := \sum_{k=1}^{K} w_k G(x; \mu_k, C_k)$
    - *What are the ways in which datum 'x' could be generated ?*
    - Estimate parameters, i.e., weights {$w_k$}, means {$\mu_k$}, covariances {$C_k$} where k=1,…,K
  - **A strategy:** maximum-likelihood estimation
    - Parameters $\theta = \{w_k, \mu_k, C_k\}_{k=1}^{K}$
    - Likelihood function: $$L(\theta|y) := P(y|\theta) := \Pi_{n=1}^{N} P(y_n|\theta) := \Pi_{n=1}^{N} \left( \sum_{k=1}^{K} w_k G(y_n; \mu_k, C_k) \right)$$
    - Maximizing log-likelihood $$\max_{\theta} L(\theta|y) = \max_{\theta} \sum_{n=1}^{N} \log \left( \sum_{k=1}^{K} w_k G(y_n; \mu_k, C_k) \right)$$
      - Log-sum cannot be simplified further

# Gaussian Mixture Model

- Fitting a GMM to data
  - No closed-form updates possible when K>1
  - Gradient-ascent optimization
    - Converges slowly in practice
  - Is there a better alternative that alleviates both above problems ?
    - Yes
    - Expectation maximization (EM) optimization algorithm
      - Iterative algorithm when closed-form updates within each iteration

“Theory without practice is empty.

Practice without theory is blind.”

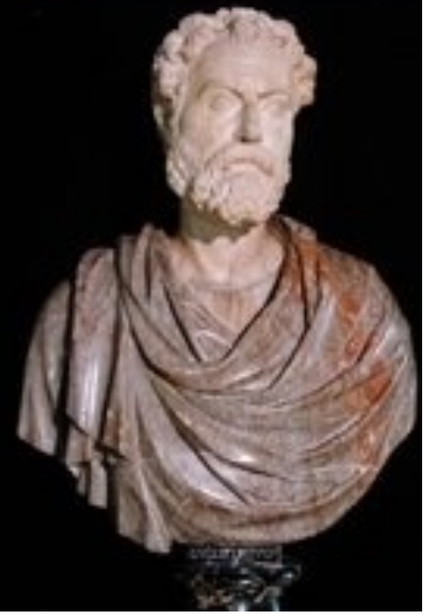
J.R. Kidd

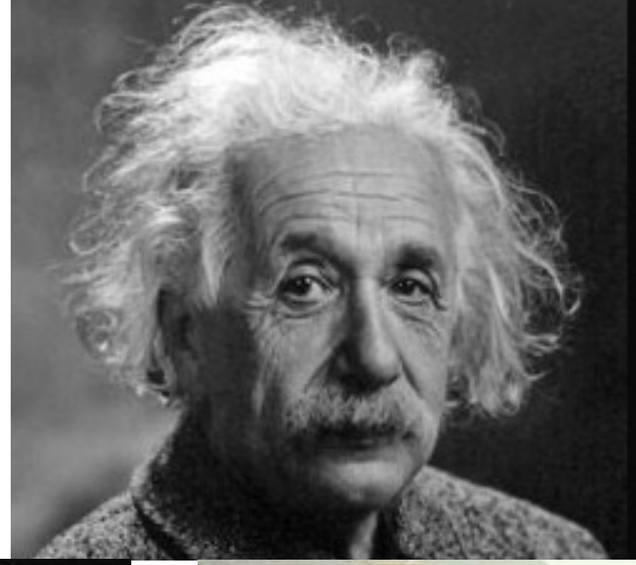
‘There is nothing more practical than a good theory’.

*Phrase attributed to Kurt Lewin, German-American psychologist, known as one of the modern pioneers of social, organizational, and applied psychology.

Practice without theory is more valuable than a theory without practice

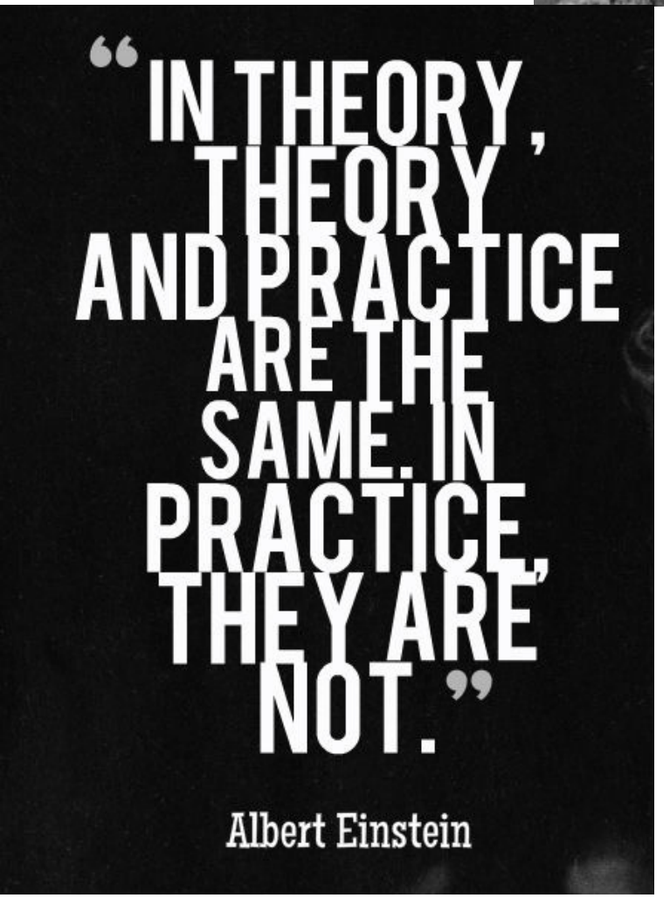
~ Marcus Quintilianus ~

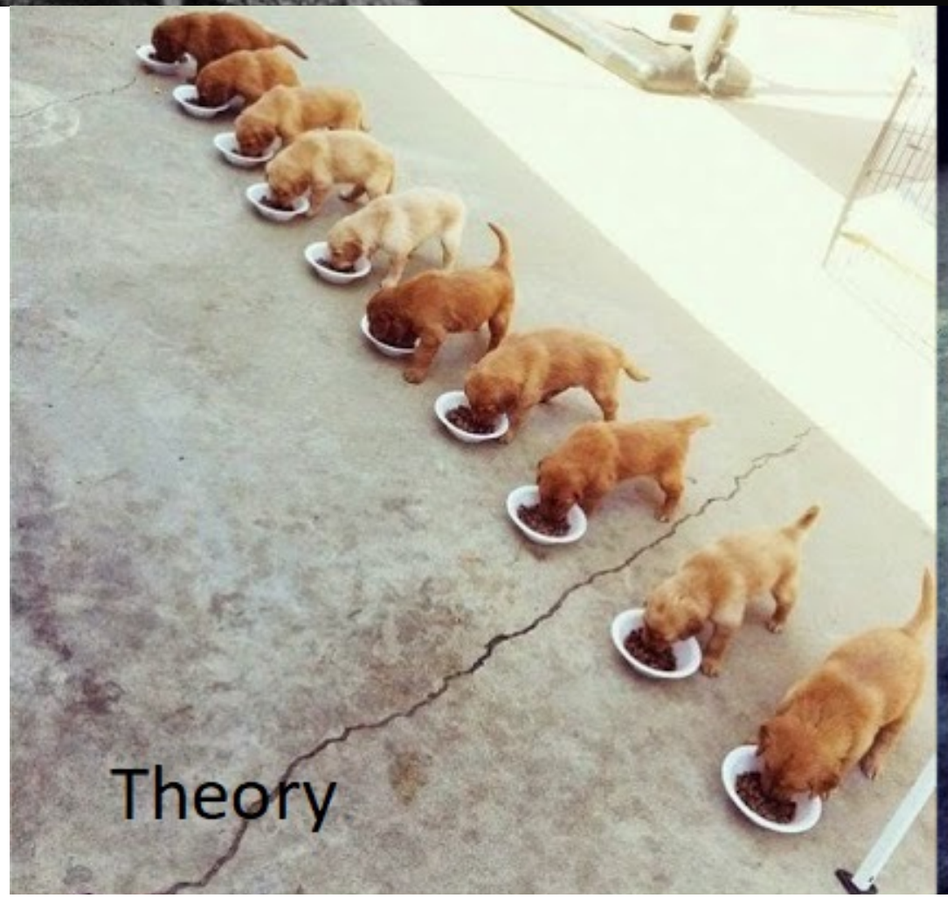




As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.

(Albert Einstein)





Theory is when you know everything but nothing works.

Practice is when everything works but no one knows why.

In our lab, theory and practice are combined: nothing works and no one knows why.

TOONY PRETZELS

# THEORY

FUNDAMENTAL OR ABSTRACT
PRINCIPLES UNDERLYING
A SCIENCE OR AN ART.

# EM Optimization

- Observed random vector Y := { $Y_n$ } where n=1,...,N
  - Leads to data y := { $y_n$ } where n=1,...,N
- Parameters: θ that needs to be estimated
- **Hidden/latent** random variables $X_n$ model missing data (for each $Y_n$)
- **Complete Data:** combination of observed + "missing data" = (X,Y) = { ($X_n$,$Y_n$) } where n=1,...,N
- Hidden variable X is designed (creatively) s.t. if missing data x were available, then it would become easy to solve for parameters
  - e.g., for segmentation, hidden variable = cluster label $X_n$ for datum $Y_n$
  - Knowing cluster-label $X_n$ values makes it easy to solve for parameters
  - Given parameters, how do we find the probability of cluster-label values ?

# EM Optimization

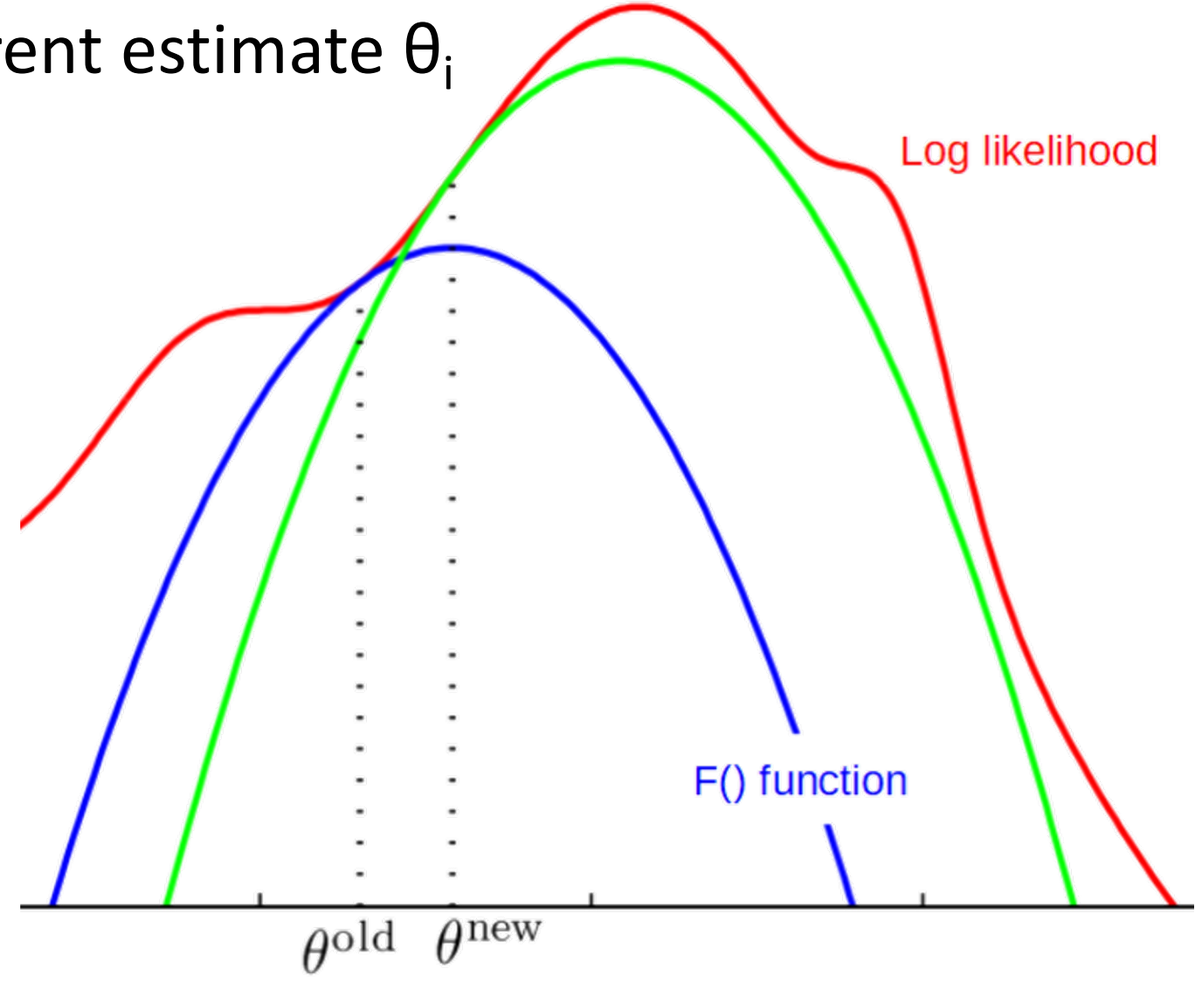
- EM optimization performs ML estimation
  - EM introduces hidden variable, then marginalizes over it
  - ML estimate is:

$$\max_{\theta} P(y|\theta) = \max_{\theta} \int_{x} P(y, x|\theta) dx$$

- What is an alternative ?
  - Treat 'x' as parameter
  - Optimize for a value of 'x'
- Why is EM preferred ?
- EM performs iterative optimization
  - Each EM iteration comprises 2 steps:
  1. E step
  2. M step

# EM Optimization

- At iteration i, parameter estimates are $\theta_i$
- E step: designs a function of parameters $\theta$, i.e., $F(\theta; \theta_i)$, that:
  1) is a lower bound for log-likelihood function
  2) touches log-likelihood function at current estimate $\theta_i$
- M step: maximizes $F(\theta; \theta_i)$ over $\theta$



# EM Optimization

March, 1951

## On Information and Sufficiency

S. Kullback, R. A. Leibler

Ann. Math. Statist. 22(1): 79-86 (March, 1951). DOI: 10.1214/aoms/1177729694

- Kullback-Leibler (KL) divergence
  - Measures dissimilarity between distributions

$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(i)\,\ln\frac{P(i)}{Q(i)} \qquad D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x)\,\ln\frac{p(x)}{q(x)}\,\mathrm{d}x$$

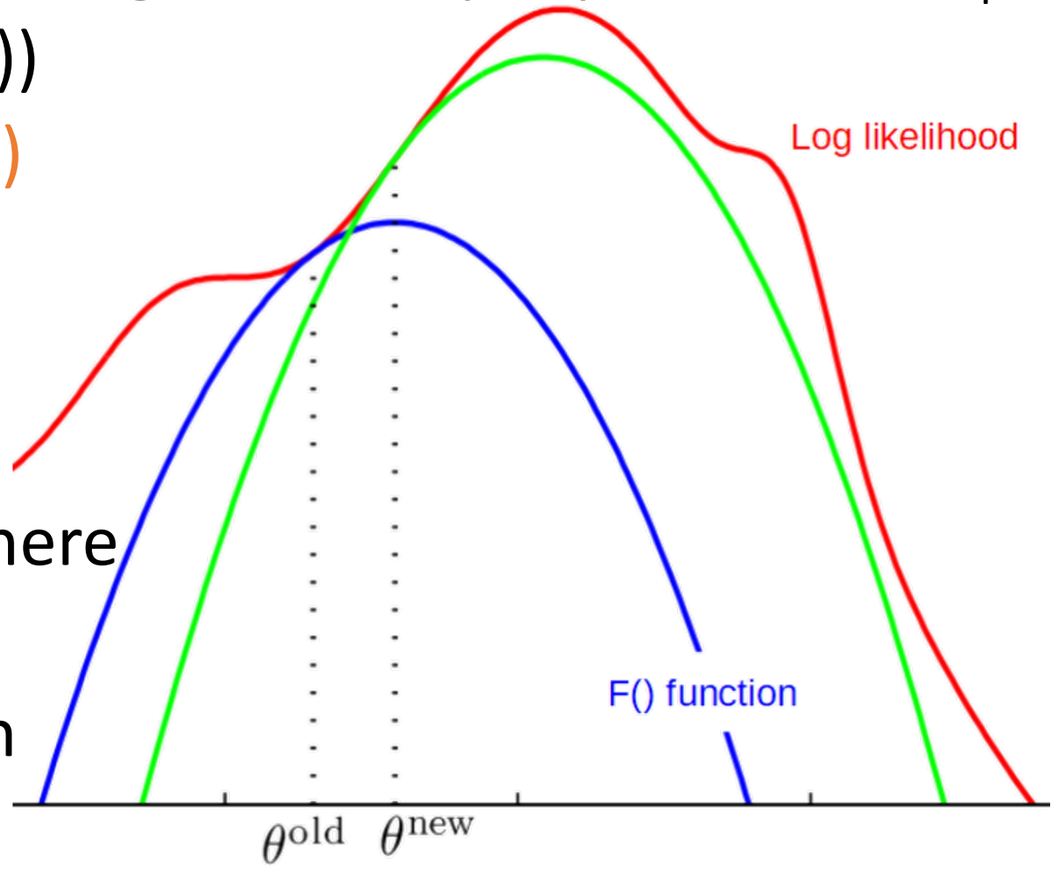
  - Expectation of log ratios of distribution values
    - Expectation taken over one of the distributions
  - Defined only when support of distribution P(.) is a subset of support of Q(.)
    - $\lim_{p(x)\to 0} p(x)\log p(x) = 0$
    - In our case (as we shall see later), support of p(.) and q(.) will be same
  - Non-negative
    - Jensen's inequality; -log(.) is convex; $E_{p(X)}[-\log(q(X)/p(X))] \geq -\log(E_{p(X)}[q(X)/p(X)]) = 0$
  - Asymmetric → NOT a distance metric. Also doesn't satisfy triangle inequality.

# EM Optimization

- Consider log likelihood $\log(P(y|\theta))$
  - Function of parameters $\theta$
- Log likelihood can be modeled as $F(q,\theta) + KL(q \,\|\, p(X|y,\theta))$ where:
  1) $F(q,\theta) := \sum_x \; q(x) \; \log ( \mathbf{P(y,x|\theta)} / q(x) )$
     - Depends on distribution **q(X)**
       - Can be considered a functional with respect to q(X)
     - Function of parameters $\theta$
  2) $KL(q \,\|\, p(X|y,\theta)) := \sum_x \; q(x) \; \log ( q(x) / P(x|y,\theta) )$
     - Non-negative $KL(q \,\|\, p) = 0$ iff $q = p$
     - Function of parameters $\theta$
- So, as function of parameters $\theta$: $F(q,\theta) = \log P(y|\theta) - KL(q \,\|\, p(X|y,\theta))$
  - So, $F(q,\theta)$ is a lower bound of $\log(P(y|\theta))$; for any chosen q(X)
  - $F(q,\theta')$ equals $\log P(y|\theta')$ iff $q(X) =$ posterior $p(X|y,\theta')$

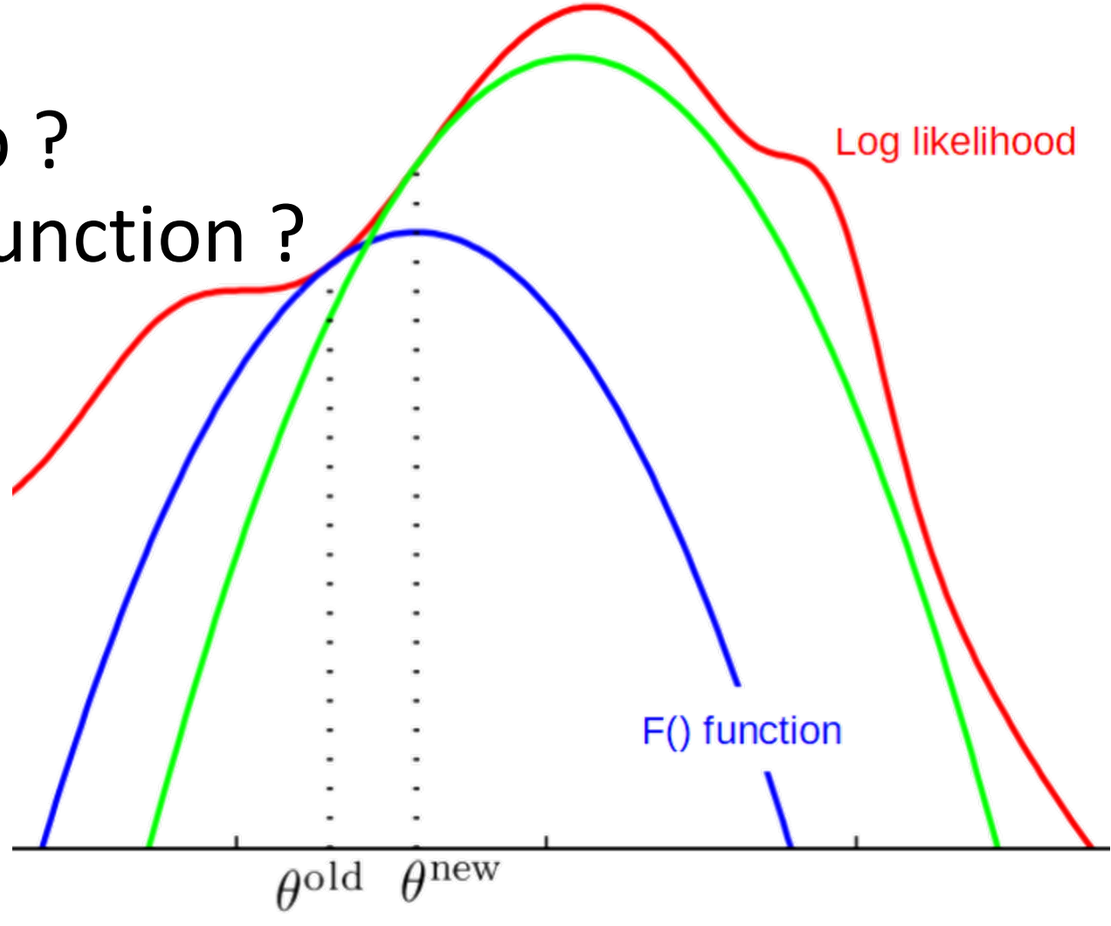
# EM Optimization

- $F(q,\theta) = \log P(y|\theta) - KL(q \| p(X|y,\theta))$
- **At iteration i**, let parameter estimates = $\boldsymbol{\theta_i}$
- **E step:** design q(.) to maximize $F(q,\theta_i)$
  - $q(X) := p(X|y,\theta_i)$ = posterior distribution of labels given data y & parameters $\theta_i$
  - So, $F(q,\theta) = \log P(y|\theta) - KL(p(X|y,\theta_i) \| p(X|y,\theta))$
  - So, evaluated at $\theta_i$, we get $F(q,\theta_i) = \log P(y|\theta_i)$
  - So, F(.) touches log-likelihood function at $\theta_i$
- **M step** = choose θ to maximize F(q,θ)
  - Observe (rewrite RHS):
    $F(q,\theta) = E_{q(X) = p(X|y,\theta i)}[\log P(X,y|\theta)] + H(q)$, where
  - H(q): isn't a function of θ; entropy of q(.)
  - Denote $E_{q(X)}[\log P(X,y|\theta)]$ as $Q(\theta;\theta_i)$ function



# EM Optimization

- Termination criterion
  - Is the graph of $Q(\theta_{i+1};\theta_i)$ increasing / non-decreasing ?
  - Between two consecutive iterations, can we terminate when relative change between parameter values ($\theta_{i+1}$ and $\theta_i$) falls below a user-defined threshold ?
- What if we don't maximize F(.) in M step ?
  What if $\theta_{i+1}$ only increases value of F(.) function ?



EM - 1977

# Maximum Likelihood from Incomplete Data via the *EM* Algorithm

By A. P. DEMPSTER, N. M. LAIRD and D. B. RUBIN

*Harvard University and Educational Testing Service*

[Read before the ROYAL STATISTICAL SOCIETY at a meeting organized by the RESEARCH SECTION on Wednesday, December 8th, 1976, Professor S. D. SILVEY in the Chair]

## SUMMARY

A broadly applicable algorithm for computing maximum likelihood estimates from incomplete data is presented at various levels of generality. Theory showing the monotone behaviour of the likelihood and convergence of the algorithm is derived. Many examples are sketched, including missing value situations, applications to grouped, censored or truncated data, finite mixture models, variance component estimation, hyperparameter estimation, iteratively reweighted least squares and factor analysis.

# EM - 1977

- Arthur P. Dempster (1929 – )
  - Joined Harvard statistics in 1957 (4 faculty in dept.)
  - Advisor: John Tukey (FFT fame) @ Princeton
    - Founding member of Princeton's statistics dept.

- NM Laird (1943 – )
  - PhD advisor: Dempster
  - Professor of Biostatistics (Emerita) at Harvard

# EM - 1977

- DB Rubin (1943 – )
  - Worked for ETS
  - In Harvard psychology grad school.
    Asked to take introductory stats courses
    because of “insufficient background”.
    Felt insulted, given many courses
    in physics @ Princeton.
    Transferred to CS.
    Eventually got PhD in stats @ Harvard.
    Rest is history.

  - 367K+ citations
  - Rubin ← Cochran ← Wishart ← Karl Pearson

# GMM Optimization via EM

- Observed random variable: Y := { $Y_n$ } for n=1,...,N
  - Data y := { $y_n$ } n=1,...,N
- PDF model: $P(\,.\,) := \sum_{k=1}^{K} w_k G(\,.\,; \mu_k, C_k)$
- Parameters: θ = { $w_k$ , $\mu_k$ , $C_k$ } where k=1,...,K
- Hidden random variable: Z := { $Z_n$ } for n=1,...,N
  - $Z_n = z_n$ is label associated with data point $y_n$
  - $z_n$ takes values 1,...,K
- Optimization strategy
  - ML estimation of parameters using EM algorithm
  - Constraint: $\sum_k w_k = 1$
- Assume, at iteration i, parameter estimates = $\theta^i$

# GMM Optimization via EM

- E step

$$Q(\theta;\theta^i) := E_{P(z|y,\theta^i)}\left[\log P(y,z|\theta)\right]$$

$$= E_{P(z|y,\theta^i)}\left[\log \Pi_n P(y_n,z_n|\theta)\right] \text{ Independent observations, Independent labels}$$

$$= E_{P(z|y,\theta^i)}\left[\log \Pi_n P(y_n|z_n,\theta)P(z_n|\theta)\right] \text{ Conditional probability}$$

$$= E_{P(z|y,\theta^i)}\left[\sum_n \log\left(P(y_n|z_n,\theta)P(z_n|\theta)\right)\right]$$

$$= \sum_n E_{P(z|y,\theta^i)}\left[\log\left(P(y_n|z_n,\theta)P(z_n|\theta)\right)\right] \text{ Linearity of expectation}$$

$$= \sum_n E_{P(z_n|y_n,\theta^i)}\left[\log\left(P(y_n|z_n,\theta)P(z_n|\theta)\right)\right] \text{ Conditional independence}$$

$$= \sum_n \sum_k P(z_n=k|y_n,\theta^i)\log\left(P(y_n|z_n=k,\theta)P(z_n=k|\theta)\right) \; Z_n \text{ is a discrete RV}$$

$$P(z|y) = \frac{P(z,y)}{P(y)}$$

$$= \prod_m \frac{P(z_m,y_m)}{P(y_m)}$$

$$= \prod_m P(z_m|y_m)$$

# GMM Optimization via EM

- E step
  - Note that, by definition, $w_k := P(z_n = k|\theta)$
  - Denote membership of datum $y_n$ in k-th cluster by: $\gamma_{nk} := P(z_n = k|y_n, \theta^i)$

$$= \frac{P(y_n|z_n = k, \theta^i)P(z_n = k|\theta^i)}{P(y_n|\theta^i)} \quad \text{Bayes rule}$$

$$= \frac{G(y_n|\mu_k^i, C_k^i)w_k^i}{\sum_k G(y_n|\mu_k^i, C_k^i)w_k^i}$$

  - Each membership is non-negative
  - For each point $y_n$, sum of memberships over K classes is 1

$$\sum_n \sum_k P(z_n = k|y_n, \theta^i) \log \left( P(y_n|z_n = k, \theta)P(z_n = k|\theta) \right)$$

# GMM Optimization via EM

- E step
  - So, $Q(\theta;\theta^i) = \sum_n \sum_k \gamma_{nk} \left( -0.5 \log |C_k| - 0.5 (y_n - \mu_k)' C_k^{-1} (y_n - \mu_k) + \log w_k \right)$
    - Mahalanobis distance, instead of Euclidean distance in k-means or FCM
- M step
  - Update parameter estimates by solving: $\arg\max_\theta Q(\theta;\theta^i)$

    under the constraint $\sum_k w_k = 1$
  - Memberships $\gamma_{nk}$ are fixed; not a function of $\theta$, but $\theta^i$
  - Solve for the Gaussian means $\mu_k$ and covariances $C_k$
    - Take partial derivatives and assign them to 0

$$\sum_n \sum_k P(z_n = k | y_n, \theta^i) \log \left( P(y_n | z_n = k, \theta) P(z_n = k | \theta) \right)$$

# GMM Optimization via EM

- M step

$$Q(\theta;\theta^i) = \sum_n \sum_k \gamma_{nk} \left( -0.5 \log |C_k| - 0.5 (y_n - \mu_k)' C_k^{-1} (y_n - \mu_k) + \log w_k \right)$$

  - Solve for **means**

$$\frac{\partial Q(\theta;\theta^i)}{\partial \mu_k} = 0 = \sum_n \gamma_{nk} C_k^{-1} (y_n - \mu_k)$$

$$\Longrightarrow \mu_k = \frac{\sum_n \gamma_{nk} y_n}{\sum_n \gamma_{nk}}$$

  - Solve for **covariances**

$$\frac{\partial Q(\theta;\theta^i)}{\partial C_k} = 0 = -\sum_n \gamma_{nk} C_k^{-T} + \sum_n \gamma_{nk} C_k^{-T} (y_n - \mu_k)(y_n - \mu_k)' C_k^{-T}$$

$$\Longrightarrow C_k = \frac{\sum_n \gamma_{nk} (y_n - \mu_k)(y_n - \mu_k)'}{\sum_n \gamma_{nk}}$$

- Derivative of a quadratic form $\partial(x^T A\, x) = x^T(A+A^T)\, \partial x$
- Derivative of a determinant $\partial|A| = |A|(A^{-T}):^T \partial A$:
- Derivative of a quadratic form involving inverse $\partial(x^T A^{-1} y) = -(A^{-T}\, x y^T\, A^{-T}):^T \partial A$:
  - where ":" corresponds to vectorization

www.ee.ic.ac.uk/hp/staff/dmb/matrix/calculus.html

# GMM Optimization via EM

$$\arg\max_{\{w_k\}} Q(\theta;\theta^i) \text{ such that } \sum_k w_k = 1$$

- M step
  - Solve for **weights** $w_k$
  - Lagrangian:

$$\arg\max_{\{w_k\}} \sum_n \sum_k \gamma_{nk} \log w_k \text{ such that } \sum_k w_k = 1$$

$$L(\{w_k\}) := \sum_n \sum_k \gamma_{nk} \log w_k + \lambda \left( \sum_k w_k - 1 \right)$$

$$= \sum_k \log w_k \gamma_k + \lambda \left( \sum_k w_k - 1 \right)$$

where $\gamma_k := \sum_n \gamma_{nk}$ is the **total membership** associated with the $k$-th Gaussian

$$\frac{\partial L(\{w_k\})}{\partial w_k} = 0 = \frac{\gamma_k}{w_k} + \lambda \implies w_k = \frac{-\gamma_k}{\lambda}$$

$$\frac{\partial L(\{w_k\})}{\partial \lambda} = 0 = \sum_k w_k - 1$$

Thus,

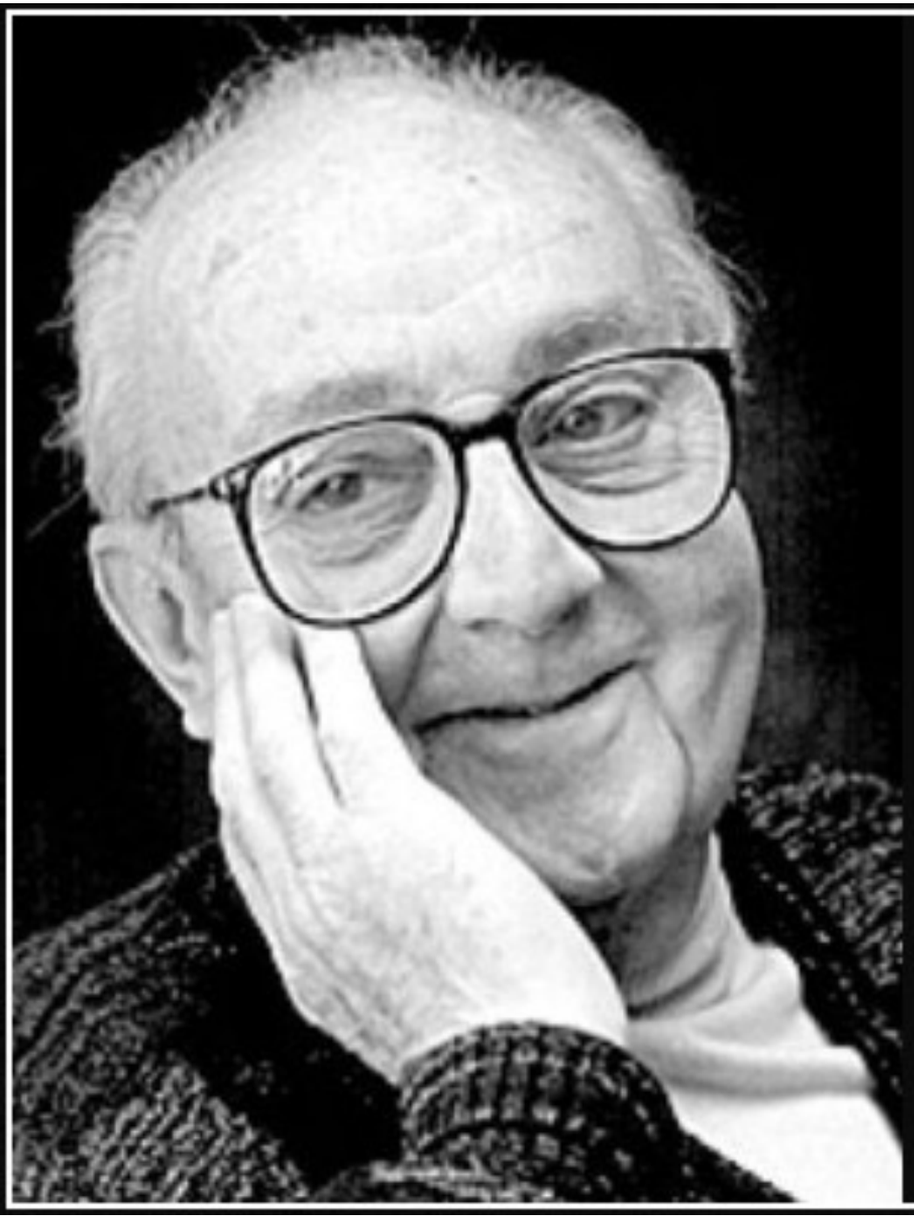
$$\lambda = -\sum_k \gamma_k \text{ and } w_k = \frac{\gamma_k}{\sum_k \gamma_k} = \frac{\gamma_k}{N}$$

where we note that $\sum_k \gamma_k = \sum_k \sum_n \gamma_{nk} = \sum_n \left( \sum_k \gamma_{nk} \right) = \sum_n (1) = N$

# GMM Optimization via EM

- Doesn't enforce spatial smoothness on segmentation
- How to do that ?
  - MRF model on label images

# George E. P. Box



## MODELS:

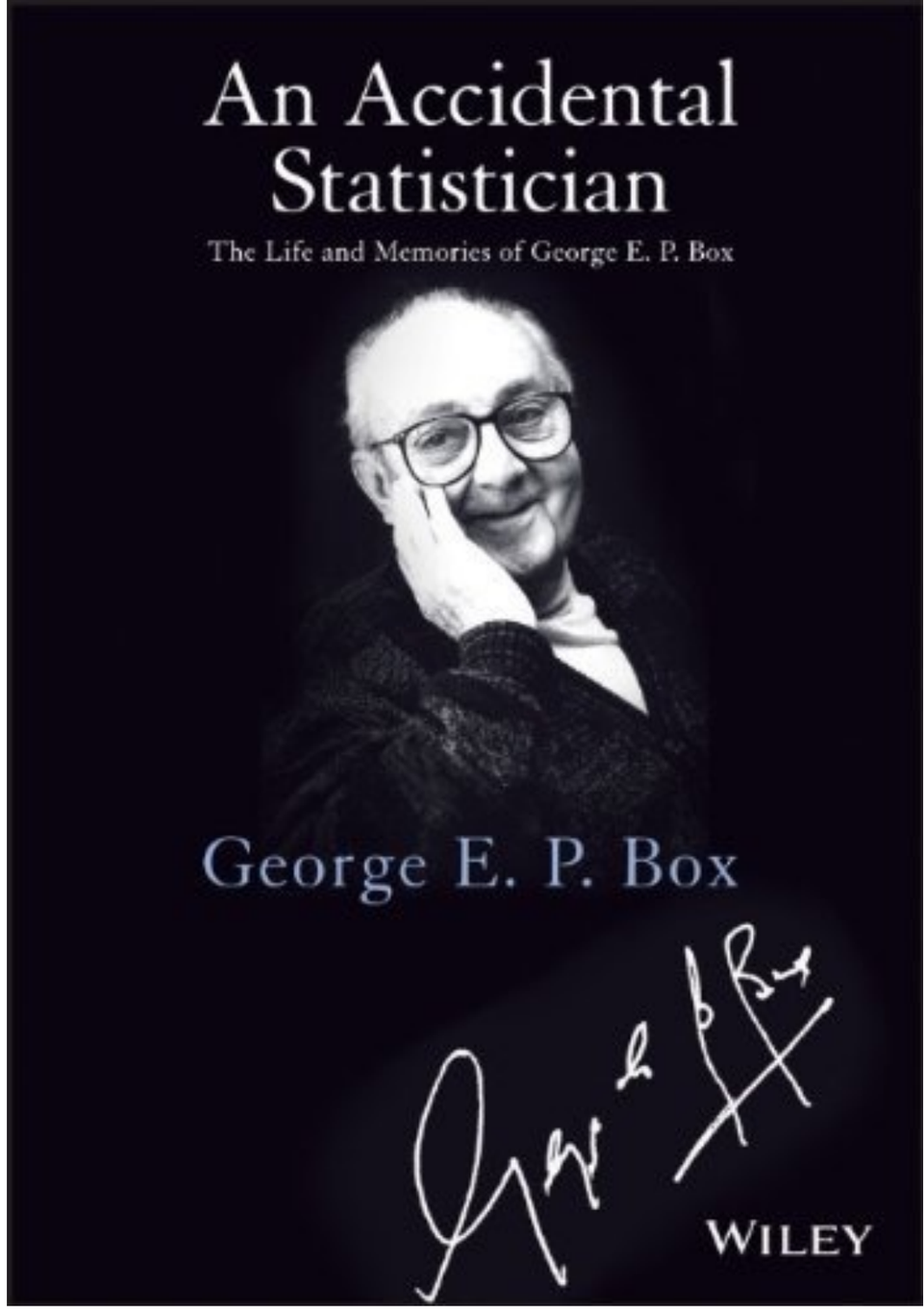
"Essentially, all models are wrong, but some are useful."

"Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful."

George E.P. Box

# George E. P. Box

- Statistician (1919 – 2013)
  - "One of the great statistical minds of the 20th century"
  - Quality control, time-series analysis, design of experiments, Bayesian inference
- Advisor: ES Pearson, son of Karl Pearson
- During World War II, performed experiments for the British Army exposing small animals to poison gas. To analyze the results of his experiments, he taught himself statistics from available texts. After war, got PhD from UCL.

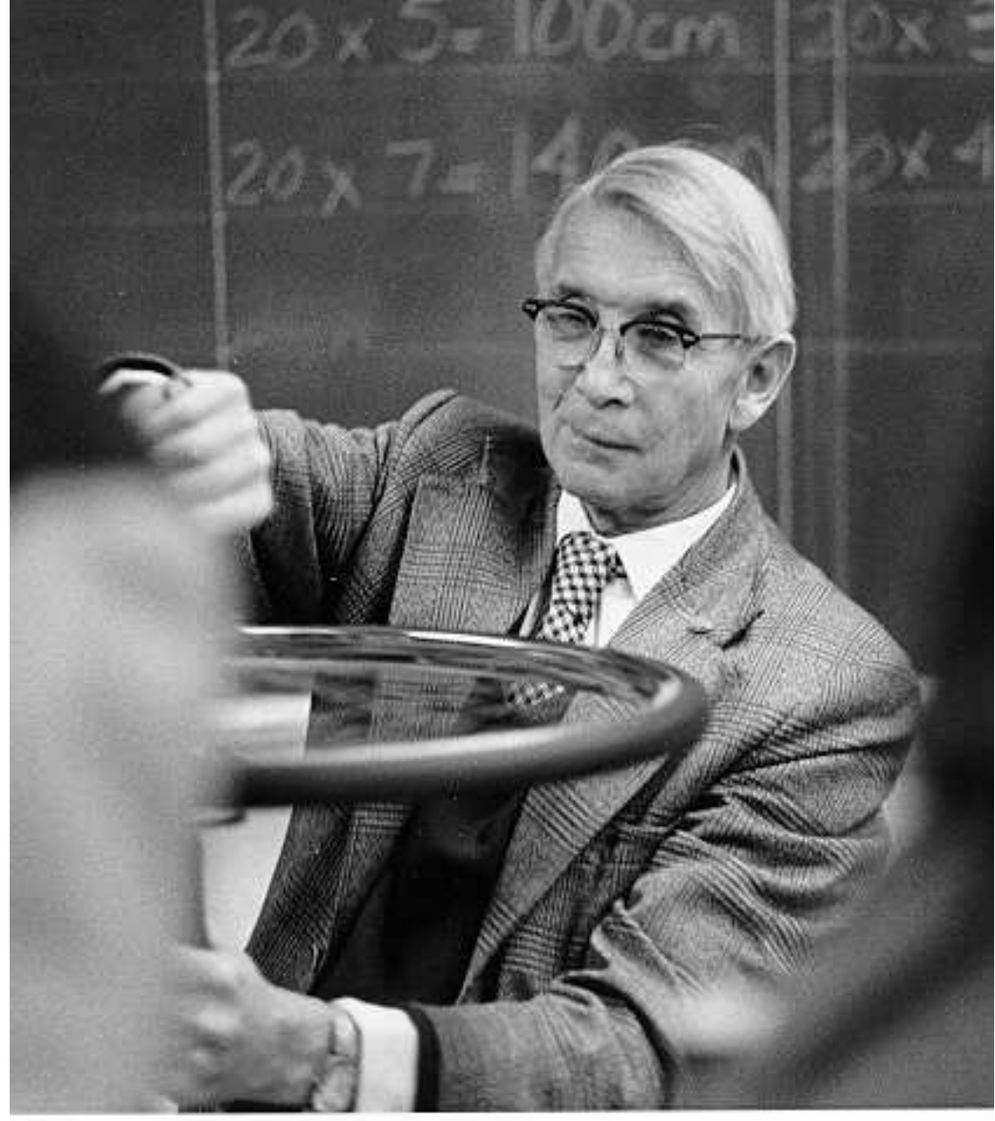


# Priors on Label Images

- **MRF on Label Images**
  - Consider a MRF where potential function is designed on 2-cliques
    - Gives a constant penalty if (neighboring) labels are unequal
    - Otherwise, gives a zero penalty
  - When labels take binary values: Ising model
  - When labels take multiple values: Potts model
  - Does a quadratic-error function make sense ?
    - Categorical random variables (not only discrete)

# Ernst Ising

- Physicist (1900 – 1998)
- Advisor: Wilhelm Lenz
  - Studied ferromagnetism with Wolfgang Pauli (Nobel Laureate)
- Finished PhD in 1924
  - PhD work = Ising model
  - Perhaps start of probabilistic graphical models
- Life
  - Forbidden to teach/research in Hitler's regime (1933+)
    - For some time, worked as a shepherd, railroad worker
  - Immigrated to US in 1947, never published later
  - Realized success of his PhD work in 1949



# Renfrey Potts

- Mathematician (1925 – 2005)
- Proposed Potts model in his PhD (1951)
- Switched fields to Operations Research
- Potts model got recognition 20 years after it was first published

# MAP-MRF Segmentation (Hard Segmentation)

- **Hidden MRF + GMM**
- Consider **label MRF X**, with labels $k \in 1,2,\cdots,K$
- Consider observed image **data Y**
- Generative model for intensities given class labels: $P(Y|X) := \Pi_i P(Y_i|X_i)$
  - e.g., for each class k, assume the model is Gaussian $P(Y_i|X_i=k) = G(Y_i|\mu_k,\sigma_k)$
- Let $\theta$ = parameters underlying noise model and MRF model (in general)
  - No weight parameters $w_k$. Why not ? Whats a 'procedure' of generating data ?
- Optimization problem: solve for segmentation x and parameter estimates $\theta$ as $\arg\max_{x,\theta} P(x)P(y|x,\theta)$
- Optimization strategy alternates between:
  1. Finding optimal parameters $\max_\theta P(y|x,\theta)$ and
  2. Finding optimal segmentation $\max_x P(y|x,\theta)P(x) = \max_x P(x|y,\theta)$

# MAP-MRF Segmentation (Hard Segmentation)

- Optimization
  - Assume MRF model parameters to be fixed
  1. Given hard segmentation x, if noise model is Gaussian, then, optimal parameters for each class = ?
     - Sample mean & sample (co)variance (over pixels labeled to that class)
  2. Given likelihood parameters, MAP segmentation is arg $\max_x$ P(x|y,θ)
     - Use iterated conditional mode (ICM) strategy
     - Rewrite P(X|y,θ)

$$= P(X_i, X_{\sim i}|y, \theta)$$

$$= P(X_i|X_{\sim i}, y, \theta)P(X_{\sim i}|y, \theta)$$ Conditional probability

$$= P(X_i|X_{N_i}, y, \theta)P(X_{\sim i}|y, \theta)$$ Markov assumption on $X$

$$= P(X_i|X_{N_i}, y_i, \theta)P(X_{\sim i}|y, \theta)$$ Conditional independence in noise model

# MAP-MRF Segmentation (Hard Segmentation)

- Optimization
  2. Given parameters, MAP segmentation is arg $\max_x P(x|y,\theta)$

$$\max_{x_i} P(X|y,\theta) = \max_{x_i} P(X_i|X_{N_i}, y_i, \theta) P(X_{\sim i}|y,\theta)$$

$$= \max_{x_i} P(X_i|X_{N_i}, y_i, \theta) \text{ Second term doesn't depend on } x_i$$

$$= \max_{x_i} P(y_i|X_i, X_{N_i}, \theta) P(X_i|X_{N_i}, \theta) / P(y_i|X_{N_i}, \theta) \text{ Bayes Rule}$$

$$= \max_{x_i} P(y_i|X_i, X_{N_i}, \theta) P(X_i|X_{N_i}, \theta) \text{ Denominator doesn't depend on } x_i$$

$$= \max_{x_i} P(y_i|X_i, \theta) P(X_i|X_{N_i}, \theta) \text{ Conditional independence assumption in}$$

- $P(y_i|X_i,\theta)$ = label likelihood
- $P(X_i|X_{Ni},\theta)$ = label prior
- How to perform this maximization ?
- Order of label updates: in parallel; update only if image probability increases

# MAP-MRF Segmentation (Soft Segmentation)

- **Hidden-MRF**
  **+ GMM**
  **+ EM**

IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL. 20, NO. 1, JANUARY 2001

## Segmentation of Brain MR Images Through a Hidden Markov Random Field Model and the Expectation-Maximization Algorithm

Yongyue Zhang*, Michael Brady, and Stephen Smith

- Number of voxels N
- Observed-data variable: Y = { $Y_i$ } where i=1,...,N
- Label MRF (hidden) variable: X = { $X_i$ } where n=1,...,N
- Number of classes = L
- Parameters θ = { $\mu_l$,$\sigma_l$ } where l=1,...,L
  - Earlier: weight parameters $w_k$ → priors on number of points derived from class k, while computing memberships
  - Now: replace weights by MRF prior
  - For voxel i, neighbors are $N_i$

$$\sum_n \sum_k P(z_n = k|y_n, \theta^i) \log \left( P(y_n|z_n = k, \theta) P(z_n = k|\theta) \right)$$

# MAP-MRF Segmentation (Soft Segmentation)

- Hidden-MRF + GMM + EM
  - Find ML estimates for parameters: $\max_\theta P(y|\theta)$, using EM
  - E step:

$$
\begin{aligned}
Q(\theta|\theta^t) &:= E_{P(X|y,\theta^t)}[\log P(X, y|\theta)] \\
&= E_{P(X|y,\theta^t)}[\log(P(y|X,\theta)P(X))] \text{ Conditional Probability} \\
&= E_{P(X|y,\theta^t)}[\log P(X) + \log \Pi_i P(y_i|X_i,\theta)] \text{ i.i.d. noise model} \\
&= E_{P(X|y,\theta^t)}[\log P(X) + \sum_i \log P(y_i|X_i,\theta)]
\end{aligned}
$$

$$
= E_{P(X|y,\theta^t)}[\overset{\log}{P}(X)] + E_{P(X|y,\theta^t)}[\sum_i \log P(y_i|X_i,\theta)] \text{ linearity of expectation}
$$

$$
= E_{P(X|y,\theta^t)}[\overset{\log}{P}(X)] + \sum_i E_{P(X|y,\theta^t)}[\log P(y_i|X_i,\theta)] \text{ linearity of expectation}
$$

$$
= E_{P(X|y,\theta^t)}[\overset{\log}{P}(X)] + \sum_i E_{P(X_i,X_{\sim i}|y,\theta^t)}[\log P(y_i|X_i,\theta)]
$$

# MAP-MRF Segmentation (Soft Segmentation)

- Hidden-MRF + GMM + EM

$$E_{P(X|y,\theta^t)}[\log P(X)] + \sum_i E_{P(X_i,X_{\sim i}|y,\theta^t)}[\log P(y_i|X_i,\theta)]$$

  - E step
    - How to handle first term ?
    - Second term requires integration w.r.t. posterior distribution on X. How to handle ?
  - Approximate expectation:

$$E_{P(x_i,x_{\sim i}|y,\theta^t)}[\log P(y_i|x_i,\theta)] \approx E_{P(x_i|x_{\sim i},y,\theta^t)}[\log P(y_i|x_i,\theta)]$$

    - Left hand side
      - Sum over possible labels at i-th voxel,
        accounting for all possibilities of neighbor labels (& their neighbors, & ...)
    - Right hand side
      - Sum over possible labels at i-th voxel, keeping rest of image *fixed*
        - This somewhat underestimates sampling variability; introduces some bias
      - But label image $X_{\sim i}$ *fixed* to what ? X is hidden
        - Fix to MAP segmentation $x_{\sim i}^{MAP}$, given current parameters $\theta^t$
        - Works well for this brain-segmentation problem

# MAP-MRF Segmentation (Soft Segmentation)

- Hidden-MRF + GMM + EM
  - E step
    - See how the MRF prior comes in

Calling $C := E_{P(x|y,\theta^t)}[\log P(x)]$ as a constant independent of $\theta$, we get

$$Q(\theta|\theta^t) - C \approx \sum_i E_{P(x_i|x^{\text{MAP}}_{\sim i},y,\theta^t)}[\log P(y_i|x_i,\theta)]$$

$$= \sum_i E_{P(x_i|x^{\text{MAP}}_{N_i},y,\theta^t)}[\log P(y_i|x_i,\theta)]$$

$$= \sum_i \sum_{l=1}^{L} P(x_i = l|x^{\text{MAP}}_{N_i}, y, \theta^t) \log P(y_i|x_i = l, \theta)$$

  - Below is what we had for GMM + EM (without MRF):

$$\sum_n \sum_k P(z_n = k|y_n, \theta^i) \log \left( P(y_n|z_n = k, \theta) P(z_n = k|\theta) \right)$$

# MAP-MRF Segmentation (Soft Segmentation)

- Hidden-MRF + GMM + EM
  - E step
    - **Memberships** = $P(x_i = l | x_{N_i}, y, \theta^t) =$

$$= P(x_i = l | y_i, \overset{\text{MAP}}{x}_{N_i}, \theta^t)$$ Conditional Independence of $X_i, Y_{\sim i}$, Given $Y_i, \overset{\text{MAP}}{X}_{N_i}$

$$= \frac{P(y_i | x_i = l, \overset{\text{MAP}}{x}_{N_i}, \theta^t) P(x_i = l | \overset{\text{MAP}}{x}_{N_i}, \theta^t)}{P(y_i | \overset{\text{MAP}}{x}_{N_i}, \theta^t)}$$ Bayes Rule

$$= \frac{G(y_i | \mu_l, \sigma_l) P(x_i = l | \overset{\text{MAP}}{x}_{N_i})}{\sum_{l=1}^{L} G(y_i | \mu_l, \sigma_l) P(x_i = l | \overset{\text{MAP}}{x}_{N_i})}$$ Conditional Independence of $Y_i, X_{N_i}$, Given $X_i$

$$\gamma_{nk} := P(z_n = k | y_n, \theta^i)$$

$$= \frac{P(y_n | z_n = k, \theta^i) P(z_n = k | \theta^i)}{P(y_n | \theta^i)}$$ Bayes rule

$$= \frac{G(y_n | \mu_k^i, C_k^i) w_k^i}{\sum_k G(y_n | \mu_k^i, C_k^i) w_k^i}$$

- Here is what we had for GMM + EM (without MRF) →

# MAP-MRF Segmentation (Soft Segmentation)

- Hidden-MRF + GMM + EM
  - E step
    - Conditional label probability is:

$$P(X_i|X_{S-\{i\}}) = \frac{\exp\left(-\sum_{a\in A_i} V_a(X_a)\right)}{\sum_{x_i'} \exp\left(-\sum_{a\in A_i} V_a(X_a)\right)} = \frac{\exp\left(-\sum_{a\in A_i} V_a(X_a)\right)}{Z_i}$$

      - Example
        - 2 classes + 4 neighborhood
        - $V(L_1,L_2) = \beta$ when $L_1$ unequal to $L_2$, where parameter $\beta > 0$; $V(L_1,L_2) = 0$ otherwise
        - At a voxel: 3 neighbors have labels A, 1 neighbor has label B
          - Conditional probability favors label A
        - At a voxel: 3 neighbors have labels B, 1 neighbor has label A
          - Conditional probability favors label B

# MAP-MRF Segmentation (Soft Segmentation)

- Hidden-MRF + GMM + EM
  - M step
    - We denote memberships by $\gamma_{nk}$

$$P(x_i = l | x_{N_i}, y, \theta^t) = \frac{G(y_i|\mu_l, \sigma_l) P(x_i = l | x_{N_i}^{MAP})}{\sum_{l=1}^{L} G(y_i|\mu_l, \sigma_l) P(x_i = l | x_{N_i}^{MAP})}$$

    - Update for cluster-mean vector and cluster-covariance matrix

$$\mu_k = \frac{\sum_n \gamma_{nk} y_n}{\sum_n \gamma_{nk}} \qquad C_k = \frac{\sum_n \gamma_{nk} (y_n - \mu_k)(y_n - \mu_k)'}{\sum_n \gamma_{nk}}$$

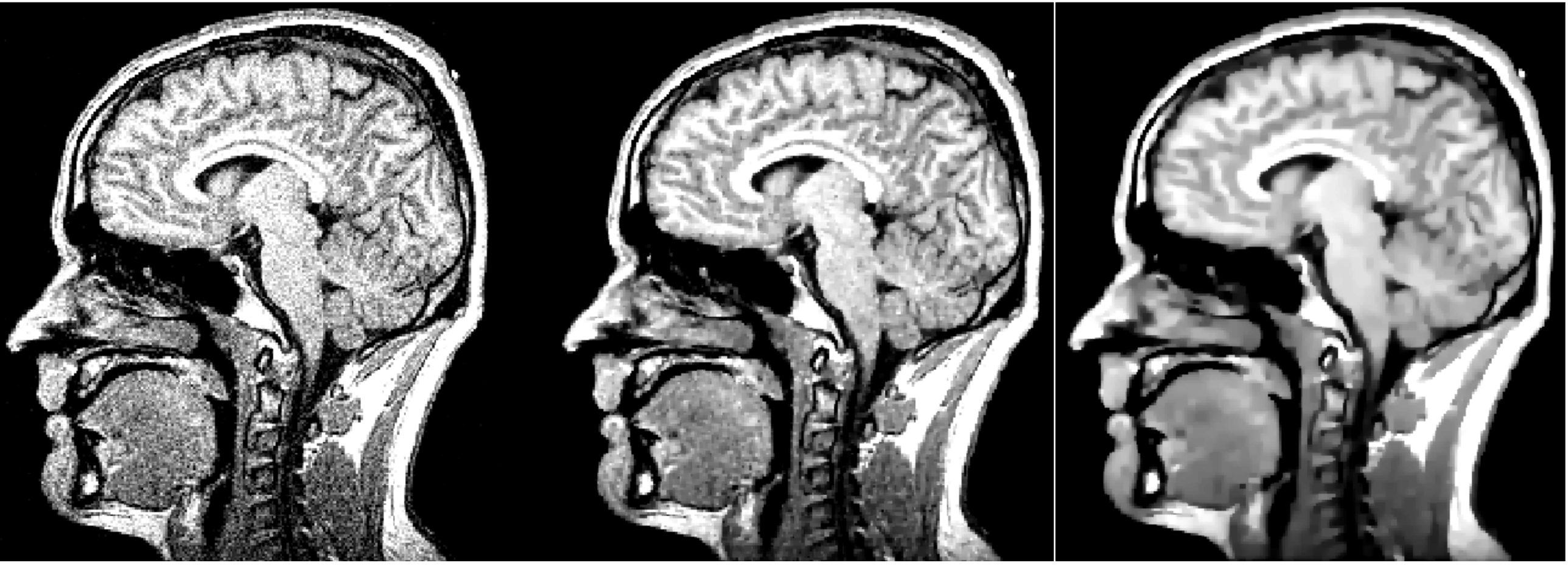
    - How will you make segmentation smoother ?
      - What parameter will you modify ?
    - What happens if the MRF prior is removed ?

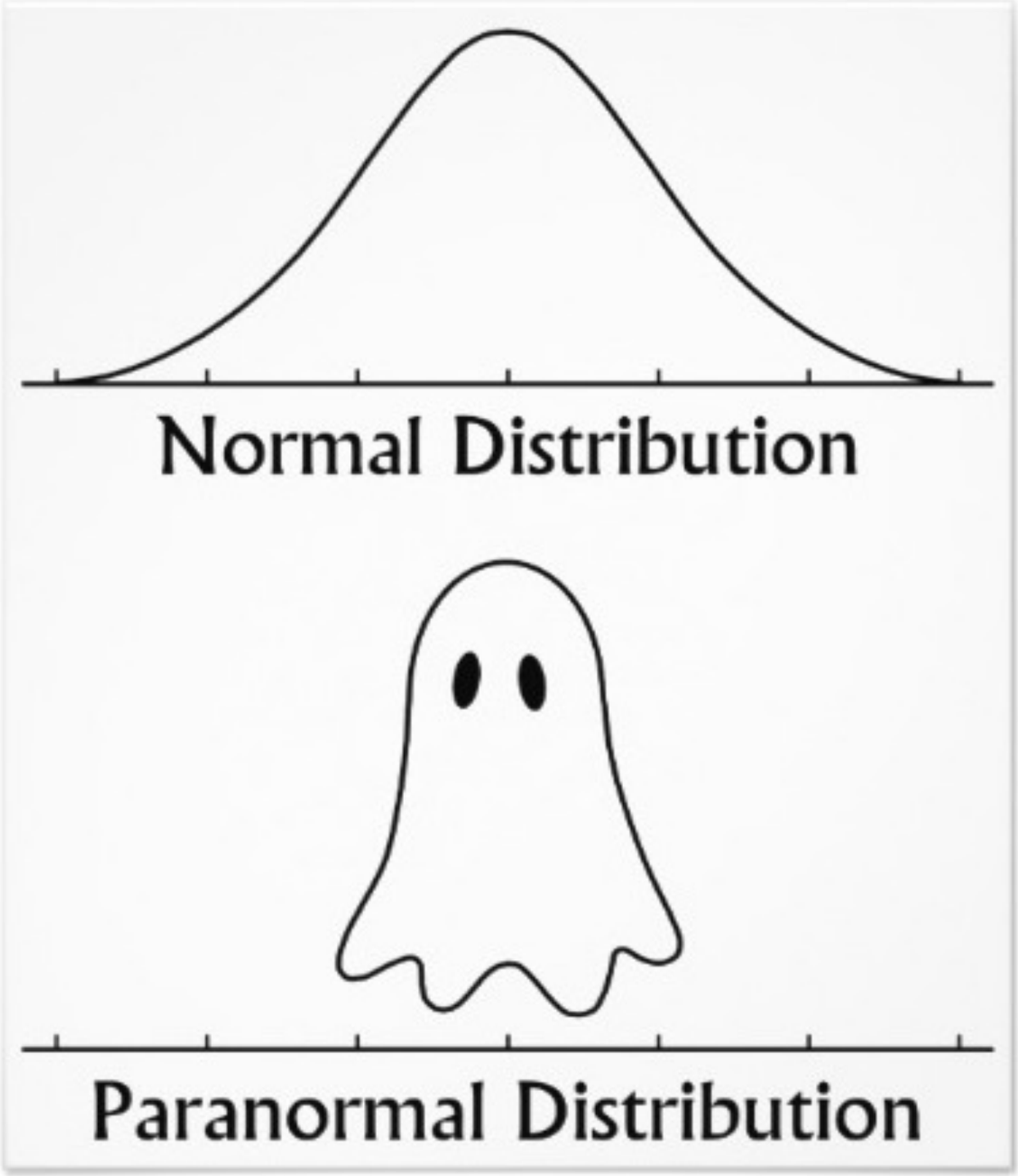
# MAP-MRF Segmentation (Soft Segmentation)

- Hidden-MRF + GMM + EM
  - Algorithm:
  - Initialize parameters: means, covariances
  - **E step (with approximation)**
    1. Compute MAP label image, given parameters
       - How to compute this ?
    2. Evaluate memberships
       - The prior will make these spatially smooth
  - **M Step**
    3. Update means and covariances, for each class
  - Repeat E and M steps, until convergence
  - **Output:** Memberships (soft, spatially smooth)
  - How to get initial segmentation ?
    - e.g., for brain tissue segmentation

- How to validate/tune an algorithm for real-world applications ?
  - Real-world phantoms
    - Ground truth is known exactly
  - Human clinical data
    - Ground truth is based on the interpretation of a set of experts
    - Intra-expert and inter-expert variability
  - Evaluation can be based on the quality of either the direct output or the output of a subsequent algorithm
- Applications:
  - Segmentation
  - Denoising
  - Reconstruction

- Examples of denoised images using different values of algorithm’s (free) parameters




Normal Distribution
Paranormal Distribution

# MAP-MRF Segmentation via s-t Cuts (Hard Seg)

- Number of voxels N
- Observed data variable = $Y = \{ Y_i \}$ where n=1,...,N
- Label MRF variable = $X = \{ X_i \}$ where i=1,...,N
- Number of classes = 2
- Let label values $x_i$ be binary, i.e., 0 or 1
- Generative model for intensities given class label $P(Y|X) := \Pi_i P(Y_i|X_i)$
- Let $\theta$ = parameters underlying noise model
  - Assume MRF (prior) model parameters to be fixed

# MAP-MRF Segmentation via s-t Cuts (Hard Seg)

- Optimization problem
  - Obtain segmentation x and parameter estimates θ by maximizing posterior
- Optimization strategy
  - Alternate between
    (1) finding optimal parameters $\max_\theta P(y|x,\theta)P(x)$ and
    (2) finding optimal segmentation $\max_x P(y|x,\theta)P(x) = \max_x P(x|y,\theta)$
- Optimizing parameters, given segmentation
  - If noise model is Gaussian,
    optimal parameters for each class are sample mean and sample variance over voxels in that class

# MAP-MRF Segmentation via s-t Cuts (Hard Seg)

- Optimizing segmentation, given parameters

$$\max_x P(x|y,\theta)$$

$$= \max_x \log P(x|y,\theta)$$

$$= \max_x \left(\log P(y|x,\theta) + \log P(x)\right)$$

$$= \max_x \left( \log \Pi_i P(y_i|x_i,\theta) + \log \frac{\exp(0.5 \sum_{(i,j)\in\mathcal{N}} \beta_{ij} V(x_i,x_j))}{Z} \right)$$

Consider only cliques of size 2

where

– $Z$ is a constant that depends only on the MRF parameters that are fixed/known

– $\beta_{ij} \geq 0$ : non-negative (convention; relative to how the potential function $V()$ is defined)

– $\beta_{ij} := \beta_{ji}$ : neighbor interactions are symmetric

– $\beta_{ii} := 0$ : interaction with self isn't allowed

# MAP-MRF Segmentation via s-t Cuts (Hard Seg)

- Assume that potential function V(·,·) is defined, for the case of binary label values, as V(a,b) := ab + (1−a)(1−b)
  - When neighbor labels are same:
    - If a=b=0 or a=b=1, then V(a,b) = 1
    - Leads to higher-prob state because $\beta_{ij} \geq 0$
  - When neighbor labels are different:
    - If a=1-b=0 or a=1-b=1, then V(a,b) = 0
    - Leads to lower-prob state because $\beta_{ij} \geq 0$

$$\max_x \left( \log \Pi_i P(y_i|x_i,\theta) + \log \frac{\exp(0.5 \sum_{(i,j)\in\mathcal{N}} \beta_{ij} V(x_i,x_j))}{Z} \right)$$

# MAP-MRF Segmentation via s-t Cuts (Hard Seg)

- Rewrite likelihood function:

$$\Pi_i P(y_i|x_i,\theta) = \Pi_i P(y_i|x_i=1,\theta)^{x_i} P(y_i|x_i=0,\theta)^{1-x_i}$$

- Simplifying objective function (log posterior)

$$\max_x P(x|y,\theta) = \max_x \left( \sum_i x_i \log P(y_i|x_i=1,\theta) + (1-x_i)\log P(y_i|x_i=0,\theta) \right.$$

$$\left. +0.5 \sum_i \sum_j \beta_{ij}(x_i x_j + (1-x_i)(1-x_j)) \right)$$

V(a,b) := ab + (1−a)(1−b)

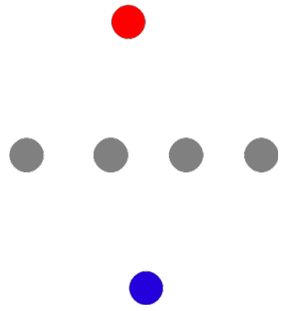
$$\max_x P(x|y,\theta) = \max_x \left( \sum_i \lambda_i x_i + 0.5 \sum_i \sum_j \beta_{ij}(2x_i x_j - x_i - x_j) \right)$$

where $\lambda_i := \log P(y_i|x_i=1,\theta) - \log P(y_i|x_i=0,\theta) = \log \frac{P(y_i|x_i=1,\theta)}{P(y_i|x_i=0,\theta)}$

- Log-likelihood ratio $\lambda_i$ is independent of (optimal) segmentation x

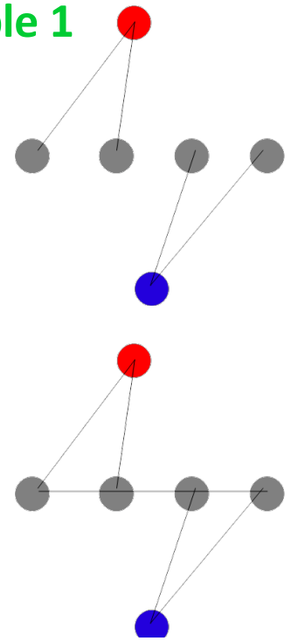
# MAP-MRF Segmentation via s-t Cuts (Hard Seg)

- Construct an s-t graph as follows:
  - Add a vertex for each voxel **i**
  - Add two additional vertices called **s** and **t**
  - For each voxel i, if $\lambda_i > 0$,
    then add an **edge** from vertex **s** to vertex **i** with cost $c_{si} := \lambda_i > 0$
  - For each voxel i, if $\lambda_i \leq 0$,
    then add an **edge** from vertex **i** to vertex **t** with cost $c_{it} := -\lambda_i \geq 0$
  - Between every pair of neighboring voxels (i,j),
    add an **edge with** cost $c_{ij} := \beta_{ij} \geq 0$

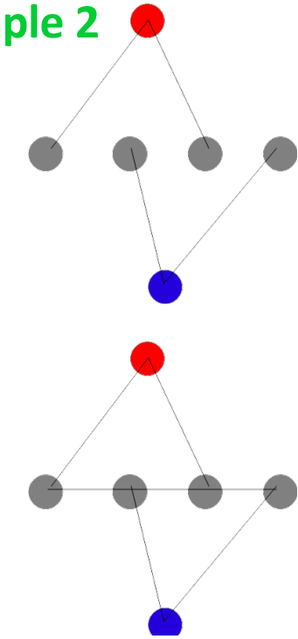


$$\log \frac{P(y_i|x_i=1,\theta)}{P(y_i|x_i=0,\theta)}$$

Example 1



Example 2



# MAP-MRF Segmentation via s-t Cuts (Hard Seg)

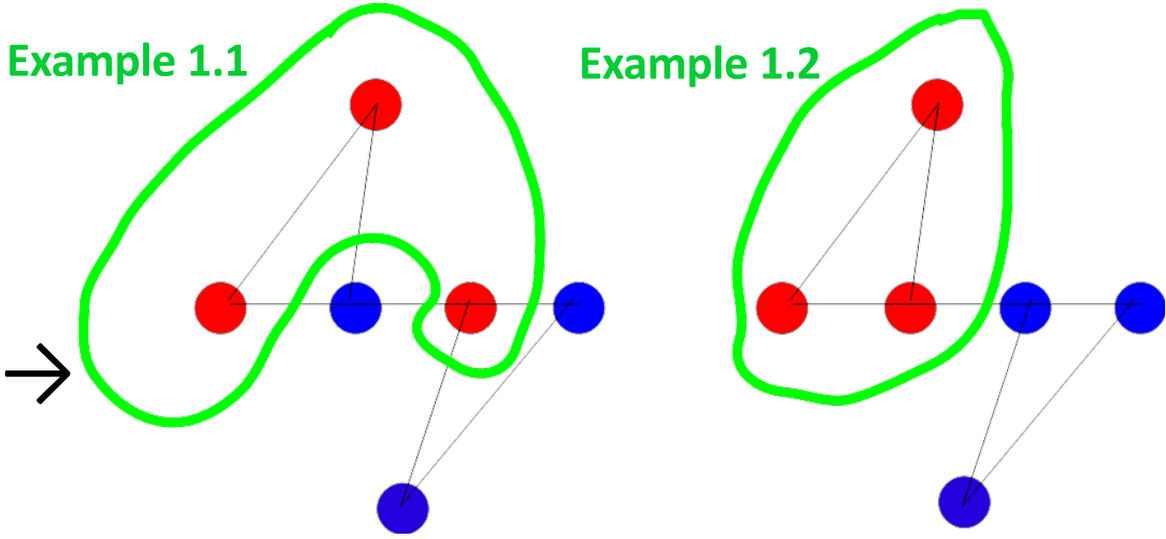
$$\log \frac{P(y_i|x_i=1,\theta)}{P(y_i|x_i=0,\theta)}$$

- A **cut** of the s-t graph is a **partition** of the graph into 2 mutually-exclusive & exhaustive sets of **vertices** s.t.
  - one set contains "s" (& all vertices in the set have label 1)
  - other set contains "t" (& all vertices in the set have label 0)

$$S := \{s\} \cup \{i : x_i = 1\}$$

$$T := \{t\} \cup \{i : x_i = 0\}$$

- Cut **separates** vertex s from vertex t
- 2 example cuts for graph in Example 1 →



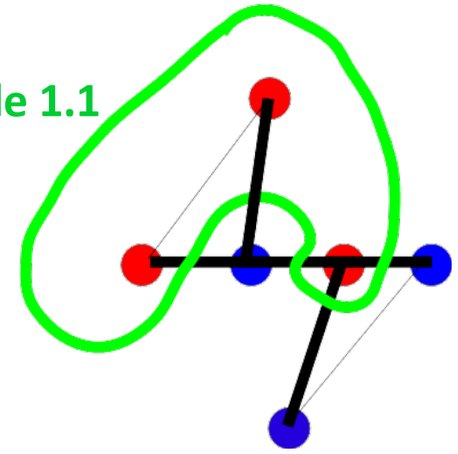

- **Capacity/cost** of cut (S,T) := sum of costs of each edge with one of the vertices in S and the other vertex in T

$$C(S,T) := \sum_{u \in S, v \in T} c_{uv}$$

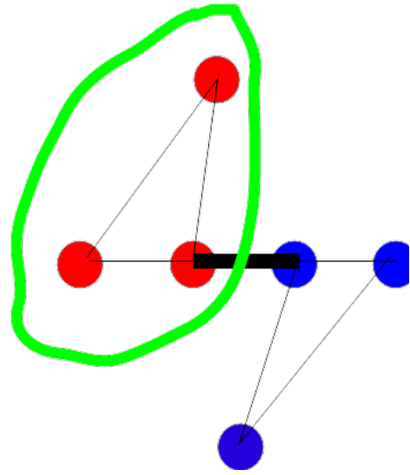
# MAP-MRF Segmentation via s-t Cuts (Hard Seg)

- **Claim: Minimum-cost s-t cut corresponds to MAP labeling**
- For some cut (S,T):
  - Or equivalently, for some labeling x:

Example 1.1



Example 1.2



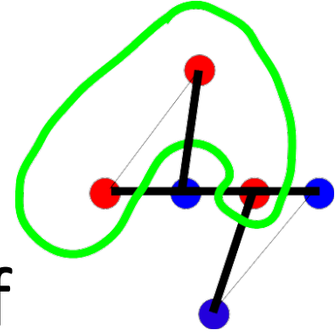
$$C(S,T) := \sum_{u \in S, v \in T} c_{uv}$$

$$= \sum_{i \in T} c_{si} \text{ Edges between vertex } s \text{ and vertices in set } T$$

$$+ \sum_{i \in S} c_{it} \text{ Edges between vertices in set } S \text{ and vertex } t$$

$$+ \sum_{i \in S-\{s\}, j \in T-\{t\}} c_{ij} \text{ Edges between vertices in set } S - \{s\} \text{ and vertices in set } T - \{t\}$$

# MAP-MRF Segmentation via s-t Cuts (Hard Seg)

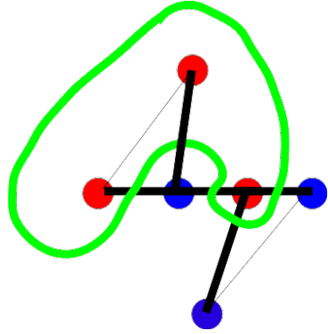


- Claim: Minimum-cost s-t cut corresponds to MAP labeling

1. In the cut, edges s–i between "s" and vertices in set T exist iff i belongs to T (i.e., label $x_i = 0$) & edge exists between s & i (i.e., $\lambda_i > 0$)
   - If edge exists, its cost = $\lambda_i$
   - For any voxel i, edge existence and cost are both modeled by $\mathbf{(1-x_i)\max(0,\lambda_i)}$
2. In the cut, edges i–t between vertices in set S and "t" exist iff i belongs to S (i.e., label $x_i = 1$) & edge exists between i & t (i.e., $\lambda_i \leq 0$)
   - If edge exists, its cost = $-\lambda_i$
   - For any voxel i, edge existence and cost are both modeled by $\mathbf{(x_i)\max(0,-\lambda_i)}$
3. In the cut, edges i–j between vertices in sets S-s and T-t exist iff labels at the two vertices are different (i.e., $x_i$ not equal to $x_j$)
   - If edge exists, its cost = $\beta_{ij}$
   - For any voxels (i,j), edge existence and cost are modeled by $\mathbf{(x_i-x_j)^2\beta_{ij}}$

# MAP-MRF Segm

- Claim: Minimum-c



$$C(S,T) := \sum_{u \in S, v \in T} c_{uv}$$

$$= \sum_{i \in T} c_{si} \text{ Edges between vertex } s \text{ and vertices in set } T$$

$$+ \sum_{i \in S} c_{it} \text{ Edges between vertices in set } S \text{ and vertex } t$$

$$+ \sum_{i \in S-\{s\}, j \in T-\{t\}} c_{ij} \text{ Edges between vertices in set } S-\{s\} \text{ and vertices in set } T-\{t\}$$

$$C(S,T) = \sum_i (1-x_i)\max(0,\lambda_i) \text{ Edges between vertex } s \text{ and vertices in set } T$$

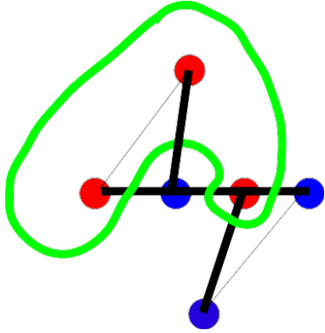
$$+ \sum_i x_i \max(0,-\lambda_i) \text{ Edges between vertices in set } S \text{ and vertex } t$$

$$+ \sum_i \sum_j 0.5(x_i - x_j)^2 \beta_{ij} \text{ Edges between vertices in set } S-\{s\} \text{ and vertices in set } T-\{t\}$$

$$= \sum_i \max(0,\lambda_i) + \sum_i x_i \Big( \max(0,-\lambda_i) - \max(0,\lambda_i) \Big) + \sum_i \sum_j 0.5(x_i + x_j - 2x_i x_j)\beta_{ij}$$

$$= \sum_i \max(0,\lambda_i) + \sum_i x_i \Big( -\lambda_i \Big) + 0.5 \sum_i \sum_j \beta_{ij}(x_i + x_j - 2x_i x_j)$$

# MAP-MRF Segmentation via s-t Cuts (Hard Seg)



- Claim: Minimum-cost s-t cut corresponds to MAP labelling
  - Proof:

$$\min_{(S,T)} C(S,T) = \min_{x} C(S(x), T(x))$$

$$= \min_{x} \left( \sum_i \max(0, \lambda_i) + \sum_i x_i \Big( -\lambda_i \Big) + 0.5 \sum_i \sum_j \beta_{ij}(x_i + x_j - 2x_i x_j) \right)$$

$$= \min_{x} \left( -\sum_i x_i \lambda_i - 0.5 \sum_i \sum_j \beta_{ij}(2x_i x_j - x_i - x_j) \right)$$

$$= \max_{x} \left( \sum_i x_i \lambda_i + 0.5 \sum_i \sum_j \beta_{ij}(2x_i x_j - x_i - x_j) \right)$$

$$= \max_{x} P(x|y,\theta)$$

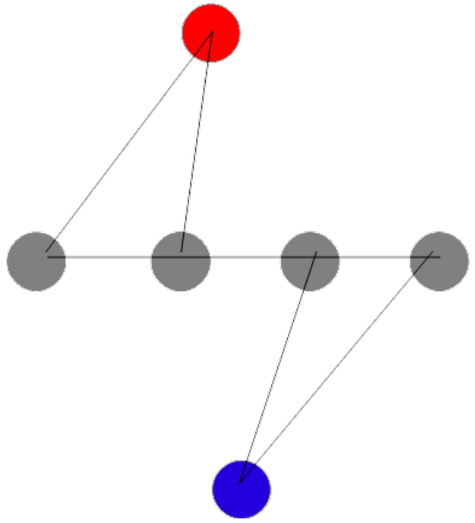
$$\max_{x} P(x|y,\theta) = \max_{x} \left( \sum_i \lambda_i x_i + 0.5 \sum_i \sum_j \beta_{ij}(2x_i x_j - x_i - x_j) \right)$$

# MAP-MRF Segmentation via s-t Cuts (Hard Seg)

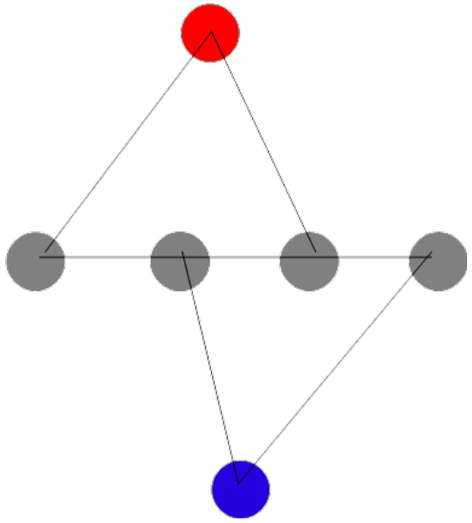
- Strengths
  - Global optimum in polynomial time
    - Various algorithms: polynomial in number of vertices or edges or both
      - http://heim.ifi.uio.no/~geird/bergen.pdf
      - http://www.cs.princeton.edu/courses/archive/spr05/cos423/lectures/07maxflow.pdf
    - NOT obvious because number of possible cuts grows exponentially with number of vertices
    - Any likelihood (noise) model can be built in
- Limitations
  - Doesn't produce soft memberships
  - Handles only 2 classes
    - Min cut with more than 2 labels is NP hard
      (but fast approximations with bounded sub-optimality exist)
      - http://www.cs.cornell.edu/courses/cs5540/2010sp/lectures/Lec18-graph-cuts.v1.pdf

# MAP-MRF Segmentation via s-t Cuts (Hard Seg)

- Example 1

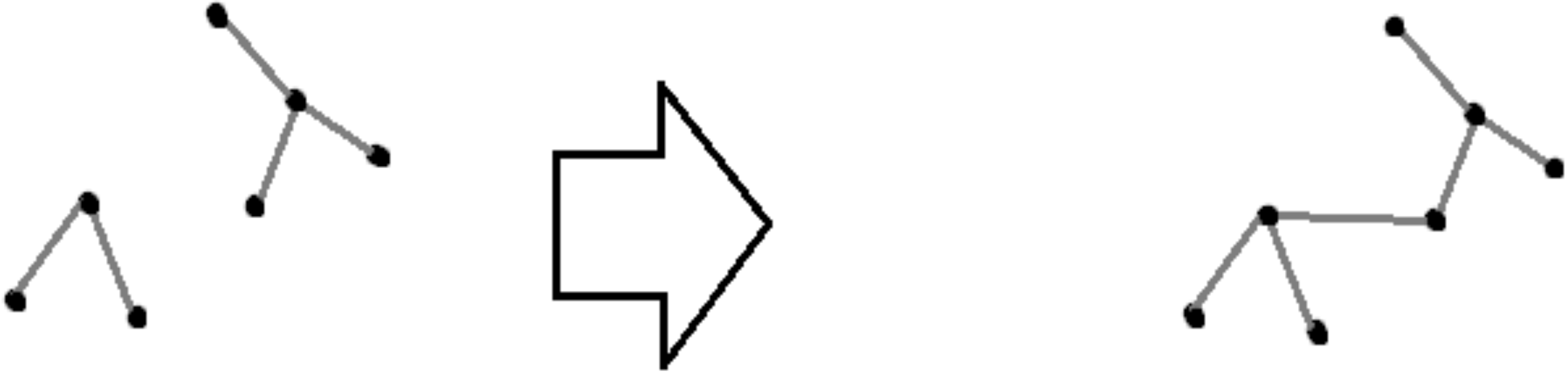


Example 2



- What are the cuts if all $\beta_{ij} = 0$ ?
- What are possible cuts if $\beta_{ij}$ values >> $|\lambda_i|$ values ?

# MAP-MRF Segmentation via s-t Cuts (Hard Seg)

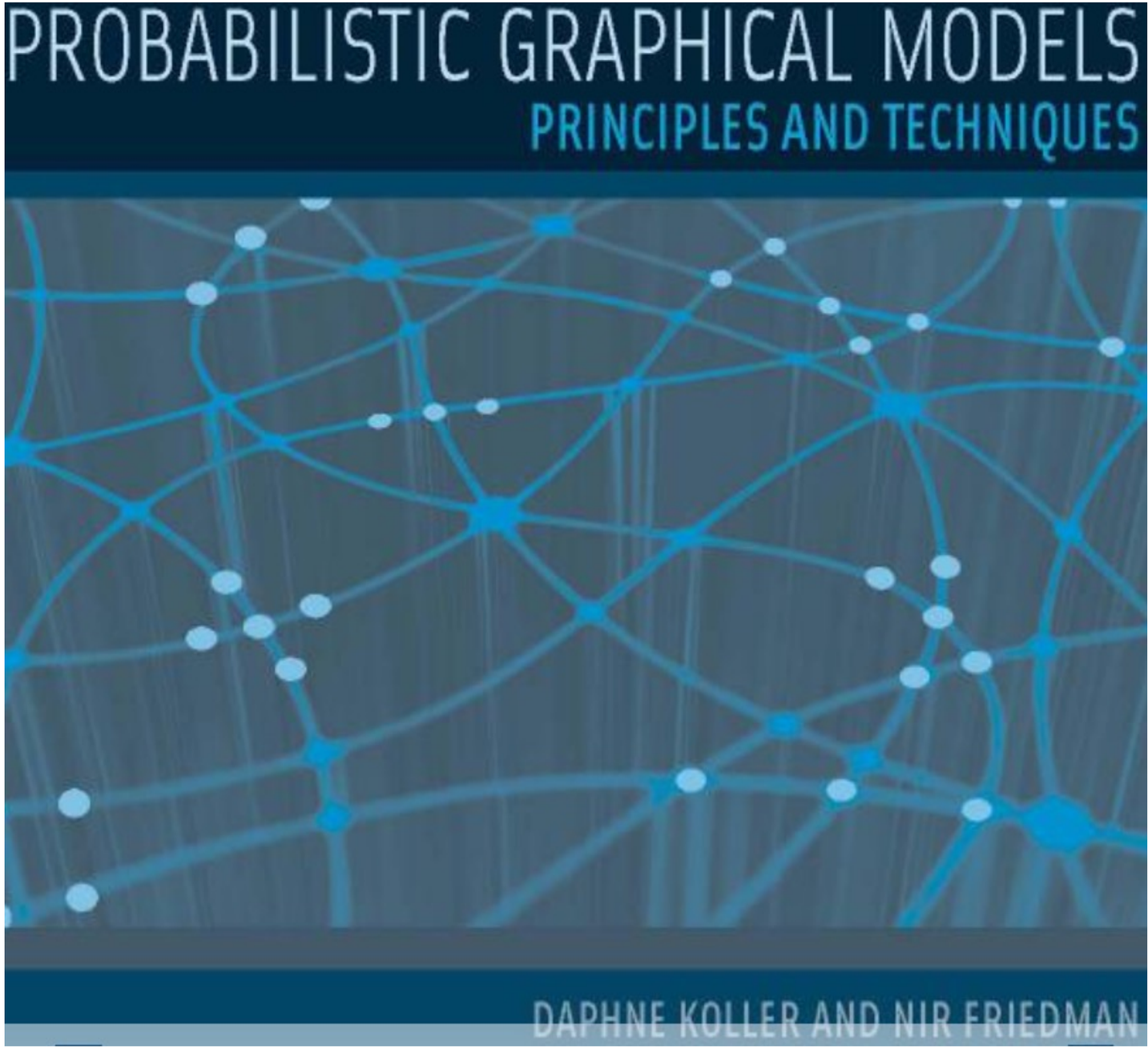


Deforestation:

When adding a branch gives you fewer trees.

# For more …


APR
Advances in Pattern Recognition
Markov
Random Field
Modeling in
Image Analysis
Stan Z. Li
Third Edition
Springer


PROBABILISTIC GRAPHICAL MODELS
PRINCIPLES AND TECHNIQUES
DAPHNE KOLLER AND NIR FRIEDMAN