Multi-Modal Data Fusion

Biomimetics and Intelligent Systems Group
Jaakko Suutala, Markus Harju

Lecture 4: Data Alignment

Outline[1]

- Spatial alignment
- Temporal alignment
- Semantic alignment
- Radiometric normalization

[1] Figures and tables adapted from the course book (H.B. Mitchell. Data Fusion: Concepts and Ideas. Springer (2012)) unless otherwise stated

Spatial Alignment

Spatial Alignment

- Conversion of local spatial positions to a common coordinate system and a common representational format

$$O =< E, \boxed{\boldsymbol{x}}, t, \boldsymbol{y}, \Delta\boldsymbol{y} >$$

- Primary fusion algorithm in many multi-modal data fusion applications
- E.g., image transformation and registration (i.e., $(x', y') = T(x, y)$ between images I_1 and I_2)
- Note: we will use symbol $\boldsymbol{u}$, x and y to represent spatial location and its coordinates (in constrast to above observation model)

Entropy

- Given a discrete random variable U with possible outcomes $u_1, u_2, \ldots, u_n$ and their probabilities $P(u_1), P(u_2), \ldots, P(u_n)$, entropy of U is given as

$$H(U) = -\sum_{i=1}^{n} P(u_i) \log P(u_i).$$

and the conditional entropy of U given V is defined as

$$H(U|V) = -\sum_{i,j}^{n} p(u_i, v_j) \log \frac{p(u_i, v_j)}{p(v_j)},$$

where $p(u_i, v_j)$ is the joint probability $U = u_i$ and $V = v_j$.

Mutual Information

- Mutual information between two random variables u and v is

$$MI(u,v) = H(u) + H(v) - H(u,v),$$

where $H(u)$ and $H(v)$ are the marginal entropies of u and v and $H(u,v)$ is their joint entropy and can be rewrite as

$$MI(u,v) = H(u) - H(u|v) = H(v) - H(v|u).$$

- Mutual information of two discrete random variables is calculated as

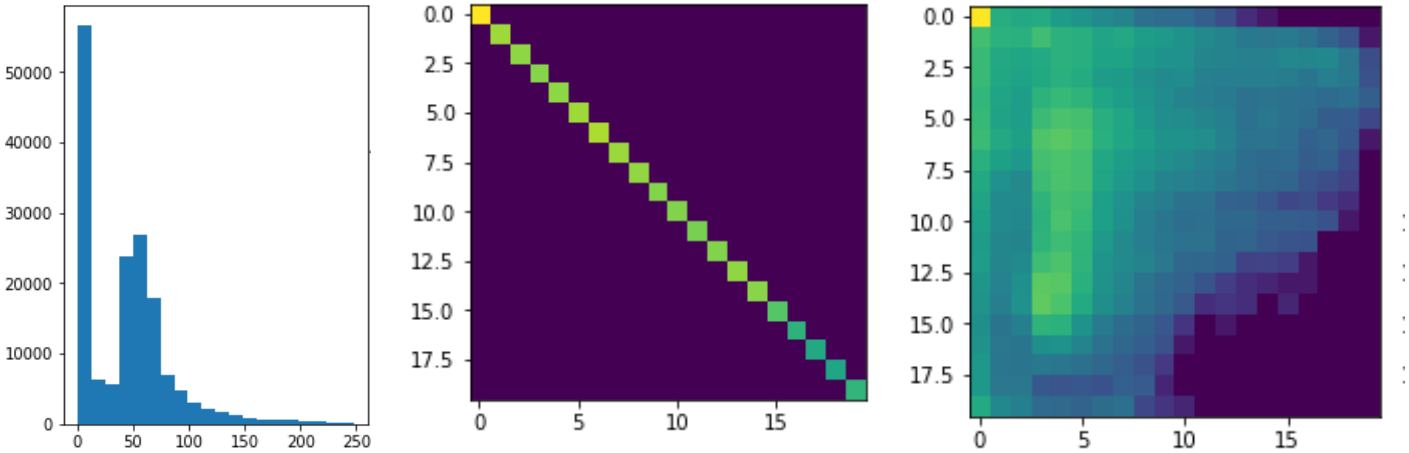
$$MI(u,v) = \sum_{v \in \mathcal{V}} \sum_{u \in \mathcal{U}} p_{UV}(u,v) \log \frac{p_{UV}(u,v)}{p_U(u) p_V(v)}$$

- In continuous domain (with differential entropy) mutual information is given as

$$MI(u,v) = \int_{\mathcal{V}} \int_{\mathcal{U}} p_{UV}(u,v) \log \frac{p_{UV}(u,v)}{p_U(u) p_V(v)} du dv$$

Mutual Information from Histogram

- Histogram is an approximation of the distribution of data
  - Counting the values occuring in certain range of values (bins)
  - 1D histogram counts observation on bins on u-axis
  - 2D histogram counts observation by the intersection of bins on the u- and v-axis



Mutual Information from Histogram (cont'd)

- Mutual information of spatial data (e.g, images) can be estimated from 2D histogram[2]
  1. Calculate 2D histogram between u and v (i.e., matrix)
  2. Normalize counts to probabilities to form the joint probability density $p_{UV}(u, v)$
  3. Form the marginal probability densities $p_U(u)$ and $p_V(v)$ by marginalizing rows and columns of normalized histogram
  4. Calculate the approximate mutual information, similar to previous slide

$$MI(u, v) \approx \sum_{v \in \mathcal{V}} \sum_{u \in \mathcal{U}} p_{UV}(u, v) \log \frac{p_{UV}(u, v)}{p_U(u) p_V(v)}$$

[2] Course book uses different formulation with bias correction

Alternative Estimation Techniques

- Kernel density estimation: a continuous alternative to histograms

$$p_U(u) = \frac{1}{NH} \sum_{i=1}^{N} K\left(\frac{u - u_i}{H}\right)$$

  where N is number of measurements, $K(u)$ is kernel (e.g., Gaussian), and H bandwidth

- Regional mutual information: using local relationships of data by stacking together the local patches of images

$$\begin{aligned} MI = &\log\left((2\pi)^{D/2} \det(\Sigma_A)^{1/2}\right) + \log\left((2\pi)^{D/2} \det(\Sigma_B)^{1/2}\right) \\ &- \log\left((2\pi)^{D/2} \det(\Sigma_J)^{1/2}\right) \end{aligned}$$

Spatial Interpolation

- In image interpolation, two dimensional continuous image $I(x, y)$ is reconstructed from the discrete pixel values $I(i, j)$, estimating the amplitude at position (x, y) from its discrete neighbours and is often modelled as impulse response $h(x, y)$

$$I(x, y) = \sum_i \sum_j I(i, j) h(x - i, y - j).$$

- Separable and symmetrical interpolation kernels

$$h(x, y) = h_x(x) h_y(y), \text{ and } h_x(x) = h_x(-x), h_y(y) = h_y(-y).$$

Name	$h(x)$
Nearest Neighbour	1 if $0 \leq \lvert x\rvert < \frac{1}{2}$; otherwise 0.
Linear	$1 - \lvert x\rvert$ if $0 \leq \lvert x\rvert < 1$; otherwise 0.
Quadratic	$\begin{cases} -2\lvert x\rvert^2 + \frac{1}{4} & \text{if } 0 \leq \lvert x\rvert < \frac{1}{2}, \\ \lvert x\rvert^2 - \frac{5}{2}\lvert x\rvert + \frac{3}{8} & \text{if } \frac{1}{2} \leq \lvert x\rvert < \frac{3}{2}, \\ 0 & \text{otherwise.} \end{cases}$
Cubic ($N = 4$)	$\begin{cases} (a+2)\lvert x\rvert^3 - (a+3)\lvert x\rvert^2 + 1 & \text{if } 0 \leq \lvert x\rvert < 1, \\ a\lvert x\rvert^3 - 5a\lvert x\rvert^2 + 8a\lvert x\rvert - 4a & \text{if } 1 \leq \lvert x\rvert < 2, \\ 0 & \text{otherwise,} \end{cases}$ where a can take the values $a = -1/2, -2/3, -3/4, -1, -4/3$.

The above formulas in the table assume x and y are given in units of the sampling interval.

Spatial Pairwise Transformation

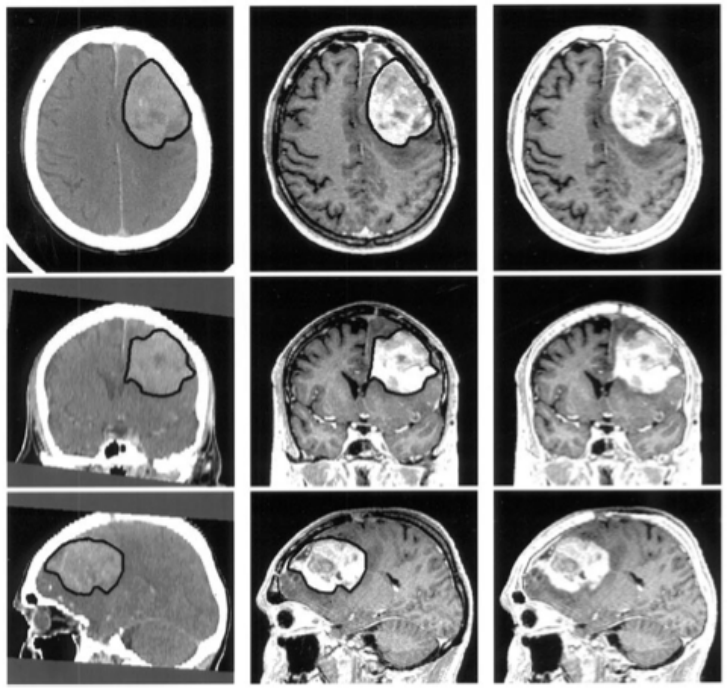
- Spatial transformation $T : (x', y') = T(x, y)$ maps a spatial location (x, y) to a new location (x', y')
- E.g., image analysis problems (remove optical distortion, register image with map coordinates, align two images etc.)
- Typical spatial (image) transformations

Name	Formula
Translation	$x' = x + a_1, y' = y + a_2.$
Similarity	$x' = a_1x + a_2y + a_3, y' = -a_2x + a_1y + a_4.$
Affine	$x' = a_1x + a_2y + a_3, y' = a_4x + a_5y + a_6.$
Perspective	$x' = (a_1x + a_2y + a_3)/(a_7x + a_8y + 1),$
	$y' = (a_4x + a_5y + a_6)/(a_7x + a_8y + 1).$
Polynomial	$x' = \sum a_{ij}x^iy^j, y' = \sum b_{ij}x^iy^j.$

- Non-rigid transformation:
  $(x'y') = T(x, y) = T_G(x, y) + T_L(x, y)$

Example: Multi-Modal Medical Image Alignment

- Spatial alignment of complementary image sources for medical analysis
- Computed tomography (CT) and magnetic resonance imaging (MRI) modalities
- Registration of images by maximizing mutual information (e.g., in the case of scale and rotation differences)



(Maes,F.,Vandermeulen,D.,Suetens,P. "Medical Image Registration Using Mutual Information". Proc. IEEE 91, 1699–1722, 2003.)

Temporal Alignment

Temporal Alignment

- Converting local sensor observation times t to a common time axis t' using the transformation $T(t)$

$$O =< E, \boldsymbol{x}, \boxed{t}, \boldsymbol{y}, \Delta\boldsymbol{y} >$$

- Basic process for creating a common representational format
- $T(t)$ is function of the position $\boldsymbol{x}$ and time t, but it is assumed that $T(t)$ is independent of $\boldsymbol{x}$
- Primary fusion algorithm in many multi-sensor data fusion applications

Dynamic Time Warping (DTW)

- Temporal alignment as a common representational format between time-series
- Can be used as a distance measure in many data mining and pattern recognition problems
- Finding *optimal* alignment between time-series $\boldsymbol{P}$ and $\boldsymbol{Q}$ minimizing the sum of the local distance $d(i,j)$ between the aligned observation pairs P_i and Q_j for euclidean distance as follows

$$d(i,j) = (P_i - Q_j)^2$$

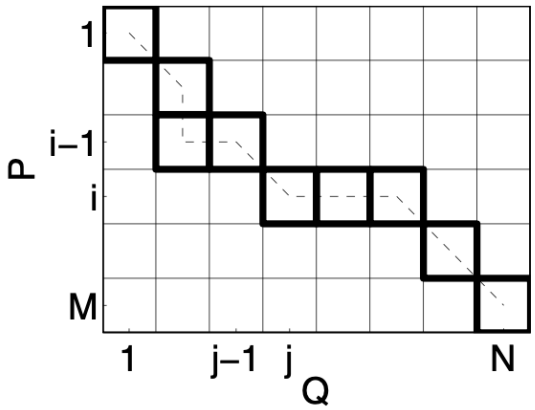
- Any other approriaty local distance measure can be used as well
- DTW distance measure can be plugged into many machine learning models (e.g., k-nearest neighbour classifier)

DTW Algorithm

- The optimal alignment is defined by means of warping path

$$\boldsymbol{W} = \{w_1, w2_2, \ldots, w_K\},$$

where
$max(M, N) \leq K \leq M + N - 1$



- Boundary conditions: $w_1 = (1, 1)$ and $w_K = (M, N)$
- Continuity: given $w_k = (a, b)$ then $w_{k-1} = (a', b')$, where $a - a' \leq 1$ and $b - b' \leq 1$
- Monoticity: given $w_k = (a, b)$ then $w_{k-1} = (a', b')$, where $a - a' \geq 0$ and $b - b' \geq 0$
- Optimal warping path

$$\boldsymbol{W}_{DTW}(\boldsymbol{P}, \boldsymbol{Q}) = \arg\min_{\boldsymbol{w}} \sum_{k=1}^{K} d(w_k)$$

DTW Algorithm (cont'd)

- Brute force search finding optimal warping path is exponential
- Using dynamic programming search can be done in $\mathcal{O}(MN)$ time with the recursion

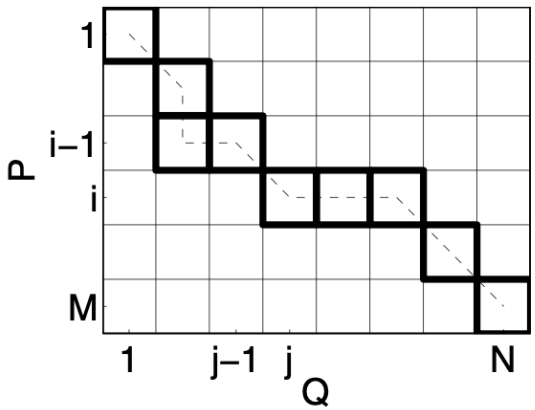
$$D(i_k, j_k) = d(i_k, j_k) + \min(D(i_k, j_k - 1), D(i_k - 1, j_k), D(i_k - 1, j_k - 1)),$$

where $D(i_k, j_k)$ is the (cumulative) cost of the optimal warping path $\boldsymbol{W}_{dtw}$ from $(1, 1)$ to (i_k, j_k) and $D(1, 1) = d(1, 1)$

DTW Algorithm (cont'd)

- The DTW algorithm build MxN matrix $D(m, n)$, $m \in \{1, 2, \ldots, M\}$, $n \in \{1, 2, \ldots, N\}$, column by column
- The recursion is started with $n = 1$ computing $D(1, 1)$ using boundary conditions $D(0, 0) = 0$, $D(0, j_K) = \infty$, and $D(i_k, 0) = \infty$
- The process is continued until the last column $n = N$ and row m = M is reached and $D(M, N)$ gives as cost of optimal aligment

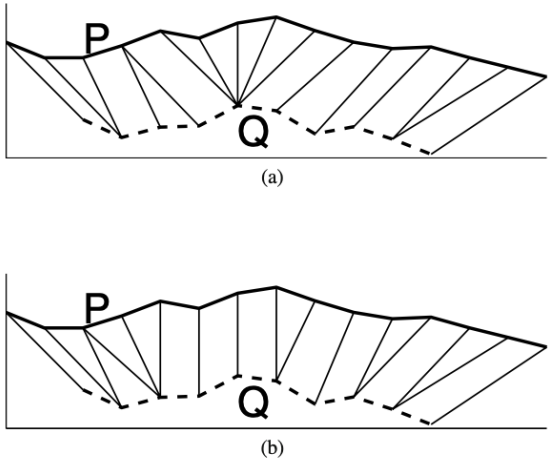
- The optimal warping path, $\boldsymbol{W}_{dtw}$, is obtained by backtracking from $D(M, N)$ to $D(1, 1)$



Extensions and Alternatives to DTW

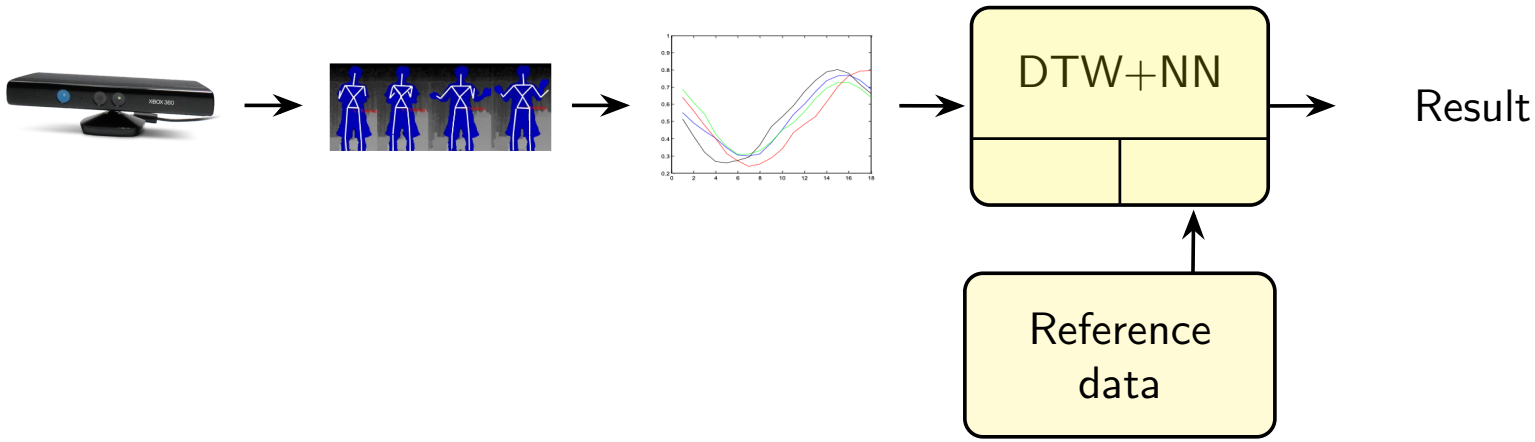
- Derivative DTW
  - Local distance $d(i,j)$ as the square of the difference between the slopes of curves $\boldsymbol{P}$ and $\boldsymbol{Q}$, i.e.,

$$d(i,j) = \left(\frac{dP}{dt}\Big|_i - \frac{dQ}{dt'}\Big|_j \right)^2$$



- Continuous DTW
  - Allows to match P_i interpolated points between Q_j and Q_{j-1} and vice verca
- One-sided DTW
  - Warping only one of the time-series $\boldsymbol{Q}$ onto the seconds time-series $\boldsymbol{P}$
  - Warping path $\boldsymbol{V} = \{v_1, v2_2, \ldots, v_k\}$, where $v_j = i$ specifies the mapping $Q_j \rightarrow P_i$

Example: DTW-based Human Action Recognition



- Estimating human pose from Kinect depth images
- Skeleton tracking of actions to time-series
- Recognizing actions using nearest neighbour (NN) classifier with DTW distance function

(S. Sempena, Nur Ulfa Maulidevi and Peb Ruswono Aryan, "Human action recognition using Dynamic Time Warping," Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, Bandung, 2011, pp. 1-5.)

Semantic Alignment

Semantic Alignment

- Conversion of multiple input data of measurements, not refering the same object to a common object or phenomena

$$O =< E, \boldsymbol{x}, t, \boxed{\boldsymbol{y}}, \Delta \boldsymbol{y} >$$

- Only semantically equivalent inputs can be fused
- Used when there are measurement from different type of sensors refering to different phenomena
- Primary fusion algorithm in many multi-sensor data fusion applications

Assignment matrix

- Label correspondence optimization problem
  - To find common representational format of sensor using different names or symbols
  - E.g., labels $\{\alpha_1, \alpha_2, \ldots, \alpha_N\}$ and $\{\beta_1, \beta_2, \ldots, \beta_N\}$ from two different clustering algorithms
- Can be solved by finding optimal assignment matrix $\tilde{\lambda}$, as follows

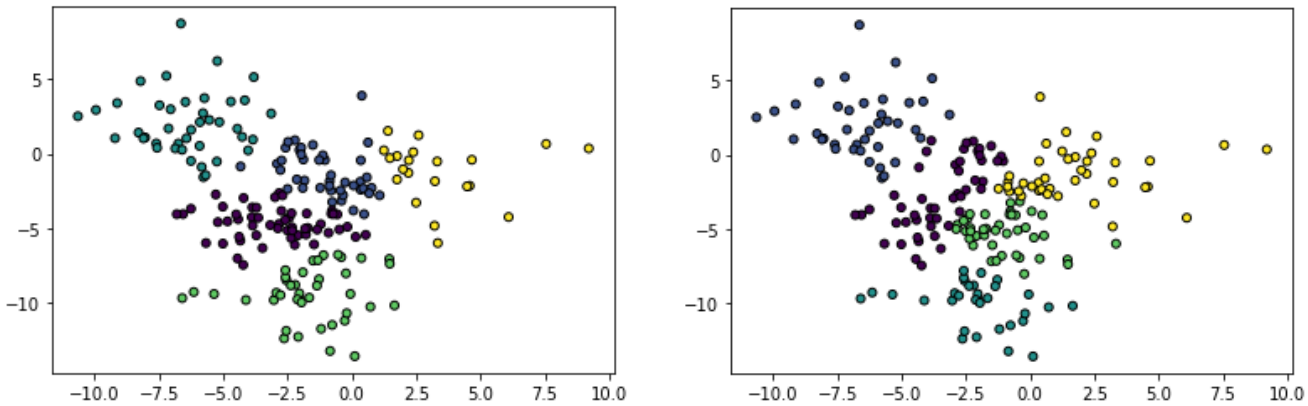
$$\tilde{\lambda} = \arg\min \sum_{k,l} C(k,l)\lambda(k,l)$$

where $C(k,l)$ is the cost of associating label α_k with label β_l and $\lambda(k,l)$ given as

$$\lambda(k,l) = \begin{cases} 1, & \text{if labels } \alpha_k \text{ and } \beta_l \text{ are associated with each other} \\ 0, & \text{otherwise} \end{cases}$$

Clustering

- Clustering algorithms are popular method for partitioning objects $O_1, O_2, \ldots, O_n$ in feature space into K classes
- Involves semantic aligment
  - Needs to perform assignments
  - Label correspondence problem with multi-sensor cluster ensembles
- Let's look at two popular algorithms more detailed: k-means (left) and spectral clustering (right)



K-means Clustering

- K-means is an iterative (unsupervised) algorithm finding K clusters in L-dimensional feature space
- The value of K is given a priori
- Let consider
  - A set of N examples/objects $\{O_1, O_2, \ldots, O_N\}$ where each object $O_n = (a_n, b_n, \ldots, d_n)^T$ is L-dimensional
  - Cluster algorithm partitioning the objects into K clusters, $P_k = (A_k, B_k \ldots, D_k)^T, k \in \{1, 2, \ldots, K\}$, in same L-dimensional feature space
  - Distance measure $d(O_n, P_k)$ between clusters $P_k = (A_k, B_k \ldots, D_k)^T$ and objects $O_n = (a_n, b_n, \ldots, d_n)^T$
- Clusters are iterative updated based on the assignments of each objects until converged

K-means Clustering (cont'd)

- If $P_k^{(r)}$ denotes the kth cluster at the rth iteration, then cluster is updated for the next iteration $P_k^{(r+1)} = (A_k^{r+1}, B_k^{r+1} \ldots, D_k^{r+1})^T$, as follows

$$A_k^{(r+1)} = A_k^{(r)} + \frac{\sum_{n=1}^{N} \lambda(n,k) a_n}{\sum_{n=1}^{N} \lambda(n,k)}$$

$$B_k^{(r+1)} = B_k^{(r)} + \frac{\sum_{n=1}^{N} \lambda(n,k) b_n}{\sum_{n=1}^{N} \lambda(n,k)}$$

$$\vdots$$

$$D_k^{(r+1)} = D_k^{(r)} + \frac{\sum_{n=1}^{N} \lambda(n,k) d_n}{\sum_{n=1}^{N} \lambda(n,k)},$$

and

$$\lambda(n,k) = \begin{cases} 1, & \text{if } P_k^{(r)} \text{ is the closest cluster to } O_n \\ 0, & \text{otherwise.} \end{cases}$$

Spectral Clustering

- Mapping observations $O_i, i \in \{1, 2, \ldots, N\}$ into K-dimensional feature space where observations are partitioned to K groups
- The value of K is given a priori
- The K-dimensional feature space is formed by extracting the eigenvectors of a normalized affinity matrix
- Algorithm:

1. Form an NxN affinity matrix $\mathcal{A}$

$$\mathcal{A}(i,j) = \exp\left(-\frac{d^2(O_i, O_j)}{\sigma^2}\right)$$

2. Set diagonal elements $\mathcal{A}(i,i), i \in \{1, 2, \ldots, N\}$ to zero

Spectral Clustering (cont'd)

3. Construct a normalized affinity matrix $\mathcal{N}$, $\mathcal{N}(i,j) = D^{-1/2}\mathcal{A}D^{-1/2}$, where

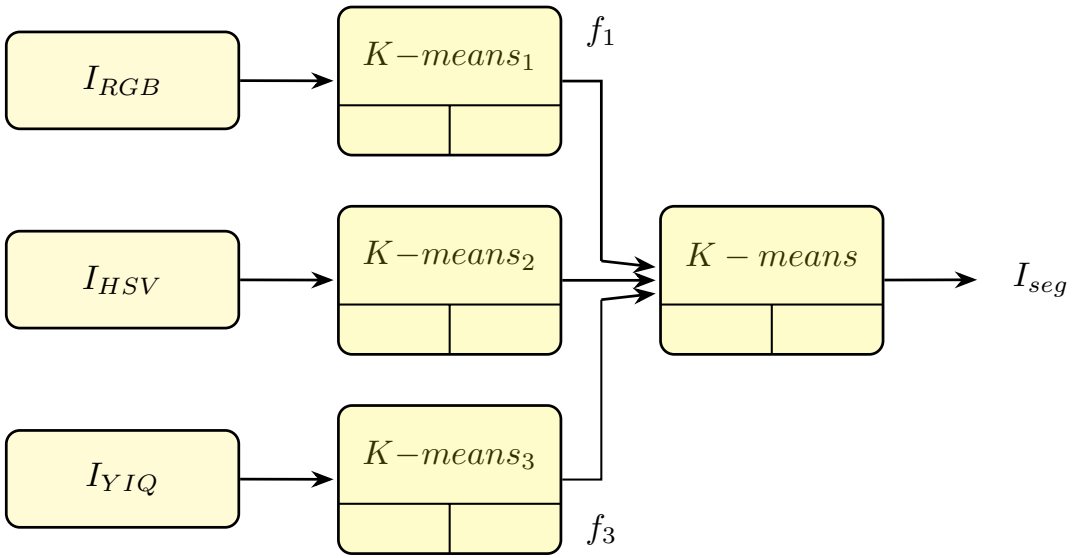
$$D(i,j) = \begin{cases} \sum_{h=1}^{N} \mathcal{A}(i,h), & \text{if } j = i \\ 0, & \text{otherwise.} \end{cases}$$

4. Calculate the eigenvector solution of $\mathcal{N}\boldsymbol{u} = \lambda\boldsymbol{u}$. If $\boldsymbol{u}_k$ is the eigenvector of $\mathcal{N}$ with the kth largest eigenvalue, then $U = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_K)$
5. Renormalize the rows of U to unit length to form a matrix V
6. Treat each row of V as a K-dimensional point. Cluster the N points with K-means algorithm
7. Assign the object O_i to the kth cluster if, and oly if, the corresponding row i of the matrix V was assigned to the kth cluster

Clustering Ensembles

- Fusing results of several clustering algorithms
  - Clustering of single sensor data with different algorithms or parameters
  - Clustering of multi-modal/multi-sensor data different or same algorithms (e.g., bagging)
- Label corresponding problem
  - Combine identity vectors $\{\boldsymbol{A}, \boldsymbol{B}, \ldots, \boldsymbol{D}\}$ from multiple clustering results into single "consensus" vector $\tilde{\boldsymbol{Y}}$
  - Solutions
    - Optimizing assigment matrices to reference algorithm
    - Concatenation: forming joint feature presentation $\boldsymbol{f}_n$ of identity vectors and fuse
    - Co-association matrix: forming binary association matrices and summing together and fusing with spectral clustering

Example: Image Segmentation with K-means Ensembles



- Color image segmentation by concatenation
  - K-means clustering of each color spaces (e.g., RGB, HSV, YIQ))
  - Form joint feature representation $\boldsymbol{f}$ of cluster identity vectors and use another k-means for final fusion

M. Mignotte, "Segmentation by Fusion of Histogram-Based K-Means Clusters in Different Color Spaces," in IEEE Transactions on Image Processing, vol. 17, no. 5, pp. 780-787, May 2008

Radiometric Normalization

Radiometric Normalization

- Conversion of sensor observations to a common scale

$$O =< E, \boldsymbol{x}, t, \boxed{\boldsymbol{y}, \Delta\boldsymbol{y}} >$$

- Semantic alignment and radiometric normalization uses similar transformations
  - But are conceptually very different (object alignment vs. measurement scaling)
- Primary fusion algorithm in many multi-sensor fusion application
- Note: we will use symbols x and y to distinguish inputs and outputs of transformation functions (in constrast to above observation model)

Scales of Measurement

- Describing the information content of variable
- Four types of scales

- Nominal scale
  - Variable values are names or labels (e.g., cat, dog, human)
  - The operations of "equal" and "inequal"
- Ordinal scale
  - Variable values with rank order (e.g., bad, ok, good)
  - Including all features of nominal scale
  - Additional operations of "greater" and "less"

Scales of Measurement (cont'd)

- Interval scale
  - Differences between numerical values can be compared (e.g., temperature in Celsius)
  - Including all features of ordinal scale
  - Can take negative values (arbitrary zero value)
  - Additional operations of "addition" and "substraction"
- Ratio scale
  - Ratios between numerical values can be compared (e.g., mass, length, energy)
  - Including all features of interval scale
  - True zero value
  - Additional operations of "multiplication" and "division"

Degree of Similarity Scales

- Type of interval scale for radiometric normalization
- A Collection of distance measures for vectorized observation
- Let have two feature vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ of M dimensions

$$\boldsymbol{A} = (a_1, a_2, \ldots, a_M), \boldsymbol{B} = (b_1, b_2, \ldots, b_M)$$

- Typical distance functions between $\boldsymbol{A}$ and $\boldsymbol{B}$

Name	Formula
Euclidean distance	$\sqrt{\sum_{m=1}^{M} \lvert a_m - b_m \rvert^2}$.
City-block distance	$\sum_{m=1}^{M} \lvert a_m - b_m \rvert$.
Chebyshev distance	$\max_m(\lvert a_m - b_m \rvert)$.
Minkowski distance	$\left[\sum_{m=1}^{M} \lvert a_m - b_m \rvert^p\right]^{1/p}$.
Canberra distance	$\sum_{m=1}^{M} \left(\lvert a_m - b_m \rvert / (\lvert a_m \rvert + \lvert b_m \rvert)\right)$.
Bray Curtis distance	$\sum_{m=1}^{M} \lvert a_m - b_m \rvert / \sum_{m=1}^{M} (a_m + b_m)$.

Degree of Similarity Scales (cont'd)

- Sometimes these feature vector are represented as histograms
- Similarity measures between the collections of bin counts
- Let have two histograms $\boldsymbol{F}$ and $\boldsymbol{G}$ of K bins

$$\boldsymbol{F} = (m_1, m_2, \ldots, m_K), \boldsymbol{G} = (n_1, n_2, \ldots, n_K)$$

- Typical similarity functions between $\boldsymbol{F}$ and $\boldsymbol{G}$

Name	Formula
L_p distance	$\left(\sum_{k=1}^{K} \lvert m_k - n_k \rvert^p\right)^{1/p}$.
χ^2 Distance	$\chi^2 = \sum_{k=1}^{K} \lvert m_k - n_k \rvert^2 / (m_k + n_k)$.
Kullback-Leibler (KL) Distance	$\sum_{k=1}^{K} \widetilde{m}_k \log(\widetilde{m}_k / \widetilde{n}_k)$, where $\widetilde{m}_k = m_k/M$, $\widetilde{n}_k = n_k/N$, $M = \sum_k m_k$ and $N = \sum_k n_k$.
Jeffrey divergence	$\sum_{k=1}^{K} \left(m_k \log(m_k/n_k) + n_k \log(n_k/m_k)\right)$.
Hausdorff Distance	$\max\left(h(f,g), h(g,f)\right)$, where $h(f,g) = \max_k \left(\min_l \left(\lvert m_k - n_l \rvert\right)\right)$.
Partial HD	$\max(h_p(f,g), h_p(g,f))$ where $h_p(f,g)$ is the p-th largest value of $\min_{k,l} \lvert m_k - n_l \rvert$.
Earth Mover's Distance	See Ex. 8.5

Radiometric Normalization Techniques

- Consider a set of logical sensor, $S_i, i \in \{1, 2, \ldots, N\}$, making N measurements x_i on an object O
- Normalizing values to common scale by changing the statistical parameters of the x_i using a parametric function f

$$y_i = f(x_i|\alpha, \beta, \gamma, \ldots, \delta)$$

where α and β are the location and scale (i.e, mean and variance) and γ, δ etc. higher order statistical parameters

- *Fixed sensor value normalization*: parameters are learned from the training samples $D = \{x_1, x_2, \ldots, x_N\}$
- *Adaptive sensor value normalization*: parameters are estimated from current measurement values $\boldsymbol{x}$

Example: Adaptive Normalization with Whitening

- Transforming a set of input measurements $x_i, i \in \{1, 2, \ldots, N\}$, into normalized value y_i, distributed according to a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i,$$

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu)^2.$$

Binarization

- Simplest radiometric normalization technique
- Given an object O, each measurements is thresholded with a local $t_n, n \in \{1, 2, \ldots, N\}$ or global t_G threshold
- The corresponding normalized measurement y_n is then given by

$$y_n = \begin{cases} 1, & \text{if } x_n \geq t_n \\ 0, & \text{otherwise} \end{cases}$$

- Threshold can be set based on some physical property, learnt on a training set D (supervised learning), or learnt on the current measurements (unsupervised learning)
- E.g., Image histrogram thresholding to detect foreground and background pixels adaptively

Parametric Normalization

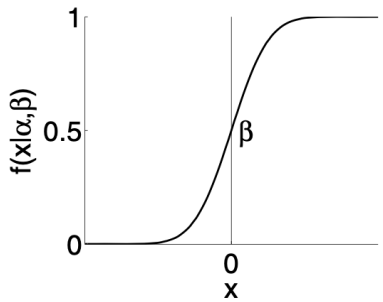
- Most common normalization technique using the parametric function

$$y_i = f(x_i|\alpha, \beta, \gamma, \ldots, \delta)$$

whose parameters $\alpha, \beta, \gamma, \ldots, \delta$ are learned from a training set D of objects O_i, $i \in \{x_1, x_2, \ldots, x_N\}$

- Psychometric transfer functions (used, e.g., in many machine learning methods and data calibration)

Name	$f(x\|\alpha,\beta)$	$df(x\|\alpha,\beta)/dx.$
Logistic	$1/(1+\exp(\alpha(\beta-x)))$	$\alpha f(x\|\alpha,\beta) \times (1-f(x\|\alpha,\beta)).$
Probit	$(\alpha/\sqrt{2\pi})\int_{-\infty}^{x}\exp-\frac{1}{2}(\alpha(t-\beta))^2 dt$	$(\alpha/\sqrt{2\pi})\exp-(\alpha(x-\beta))^2/2).$
Gumbel	$1-\exp(-\exp(\alpha\ln(x)-\alpha\ln(\beta)))$	$(\alpha/x)(1-f(x\|\alpha,\beta))\exp(\alpha\ln(x)-\alpha\ln(\beta)).$
Weibull	$1-\exp(-(x/\beta)^{\alpha})$	$\alpha x^{\alpha-1}(1-f(x\|\alpha,\beta))/\beta^{\alpha}.$
Quick	$1-2^{-(x/\beta)^{\alpha}}$	$\ln(2)\alpha x^{\alpha-1}(1-f(x\|\alpha,\beta))/\beta^{\alpha}.$



Parametric Normalization (cont'd)

- Commonly used parametric normalization approaches (e.g, in feature normalization)
  - Min-max scaling: $y = (x - \min_i(x_i))/(\max_i(x_i) - \min_i(x_i))$ (retains the input distribution)
  - Z-Transform: $y = (x - \mu)/\sigma$, where μ and σ are sample mean and standard deviation
- Summary of the approaches

Function	Formula
Min-Max	$y = (x-a)/(b-a)$, where $a = \min_i(x_i)$, $b = \max_i(x_i)$. In trimmed min-max we replace, respectively, a and b by the lth smallest and largest x_i values. The resulting normalized value is $y = \min(\max(0,(a-x)/(b-a)),1)$. *Note*: Only min-max values y_i retains the same distribution as the input x_i values.
Z-Transform	$y = (x-\mu)/\sigma$, where $\mu = \sum_k x_k/K$, $\sigma^2 = \sum_k (x_k-\mu)^2/(K-1)$. In robust Z-transform we replace μ by median$\{x_i\}$ and Q by the interquartile distance $(x_{(3N/4)} - x_{(N/4)})/2$, where $x_{(l)}$ is the lth largest value in $\{x_i\}$.
Robust Tanh	$y = \frac{1}{2}\tanh(\alpha(x-\mu_H)/\sigma_H + 1)$, where α determines the spread of the normalized scores and μ_H and σ_H are robust Hampel mean and standard deviation estimates of the $x_i, i \in \{1,2,\ldots,N\}$ [22].

Ranking

- Robust normalization technique for adaptive score normalization
- Consider input data of N objects $O_n, n \in \{1, 2, \ldots, N\}$, each with measurement x_n, corresponding rank r_n is

$$r_n = m \text{ if } x_n \text{ is } m\text{th smallest value}$$

- If there are ties among the x_n, same rank is set

Example: Ranking Fusion of PCA and LDA

- Example: rank fusion of PCA and LDA (see also Lecture 3, pages 19-20)
- Let d_n and D_n denote the Euclidean distance between test image y and training images $Y_n \in \{1, 2, \ldots, N\}$ projected onto PCA and LDA subspaces. The distance can be converted to ranks

$$r_n = k \text{ if } d_n \text{ is } k\text{th smallest PCA distance}$$

$$R_n = k \text{ if } D_n \text{ is } k\text{th smallest LDA distance}$$

- Rankings can be then fused together

$$F_n = r_n + R_n$$

- Test image is classified as belonging to the n^*th training image

$$n^* = \arg\min(F_1, F2, \ldots, F_N).$$

Conversion to Probabilities

- Normalization of output y to *a posterior* probability
- Function $f(x)$ approximating the *a posterior* probability density function $p(C = c_k|\boldsymbol{x}, I)$
- E.g. non-Bayesian classifiers not producing or estimating the output probability directly
- Typical conversion functions

Method	Description
Platt calibration	Model $f(x)$ using a sigmoid function.
Histogram binning	Approximate $f(x)$ using a discrete histogram.
Kernel density estimation	Approximate $f(x)$ using a kernel density function, i. e. a continuous bin histogram.
Isotonic Regression	Model $f(x)$ using an isotonic function, i. e. a function of unknown shape which is non-decreasing.

Conversion to Probabilities (cont'd)

- Example: converting support vector machine (SVM) scores into probability estimates using Platt calibration
  - SVM produces non-calibrated distance from the classification decision boundary
- Parametric logistic function can be used to convert these distances into a estimate of class probabilities (binary classification case), as follows

$$f(x|\alpha,\beta) = \frac{1}{1+\exp(\alpha(\beta - x))},$$

where x is the output distance/score and α and β are the location and scale parameters learned from the training data. Function gives an estimated *a posterior* probability of example/object belonging to class $C = c_2$

Conversion to Probabilities (cont'd)

- The optimization of α and β can be by minimizing the negative log likelihood of the data set D

$$(\alpha, \beta) = \arg\min \left[-\sum_{i=1}^{N} (y_i \ln f(y_i|\alpha, \beta) + (1 - y_i) \ln(1 - f(y_i|\alpha, \beta))) \right]$$

where y_i is the binary class label of ith example in the dataset D.

- In the case of unbalanced dataset to prevent overfitting, the modified class labels $\tilde{y}_i$ can be used in minimizaiton procedure, where

$$\tilde{y}_i = \begin{cases} 1/(N_1 + 1), & \text{if } O_i \text{belongs to class } C = c_1 \\ (N_2 + 1)/(N_2 + 2), & \text{if } O_i \text{belongs to class } C = c_2 \end{cases}$$

and N_1 and N_2 are the number of examples/objects belonging to each class.