UNIVERSITY OF OULU

Multi-Modal Data Fusion

Biomimetics and Intelligent Systems Group
Jaakko Suutala, Markus Harju

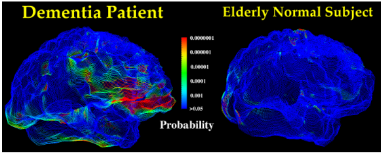
Lecture 3: Common representations

Outline

- Common coordinate systems
- Spatio-temporal transformation
- Subspace methods (PCA, LDA)
- Multiple training sets

Common coordinate systems

- a fundamental task in data fusion is to convert the sensor observations to common format
- this makes the sensor observations compatible for fusion process
- An example: brain atlas is standardized anatomically based 3D coordinate system for brain images
  - all brain have same size and orientation in new coordinate system
  - enables voxel-by-voxel comparison
  - allows for automatic labeling of structures in patient scans



(Figure: Thompson et. al: Brain image analysis and atlas construction, SPIE Press, 2000)

Common coordinate systems

Recall the sensor observations from Lecture 2 as

$$O = < E, \mathrm{x}, t, \mathrm{y}, \Delta \mathrm{y} > .$$

This gives rise to the following functions for conversion to common representational format:

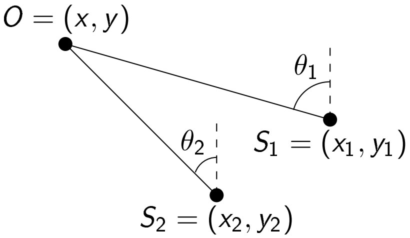
- Spatial alignment: local positions x are transformed to common coordinate system
- Temporal alignment: local times t are transformed to common time axis. E.g. dynamic time warping.
- Semantic alignment: multiple inputs are transformed so that they refer to same object/phenomena.
- Radiometric normalization: sensor values y and their uncertainties Δy are normalized to common scale.

More about these in Lecture 4.

Typically the construction of common coordinate system is the primary fusion algorithm.

Common coordinate system: Example 1

Problem: estimate location of an object O from bearing (angular) measurements at sensor locations $S_m, m = 1, 2, \ldots, M$.



Measured bearing:

$$\theta_m = \phi_m + w_m, \quad w_m \sim N(0, \sigma_m^2) \text{ i.i.d.}$$

True bearing:

$$\phi_m = \arctan \frac{x - x_m}{y - y_m}, \quad m = 1, 2, \ldots, M$$

Common coordinate system: Example 1

A posteriori probability density is

$$p(\theta|\phi, I) = \exp\left[-\sum_m \frac{1}{2\sigma_m^2}\left(\theta_m - \arctan\frac{x - x_m}{y - y_m}\right)^2\right] / \prod_m \sigma_m\sqrt{2\pi}$$

Using Bayes' theorem we obtain

$$p(x, y|\theta) \sim \pi(x, y|I)\exp\left[-\sum_m \frac{1}{2\sigma_m^2}\left(\theta_m - \arctan\frac{x - x_m}{y - y_m}\right)^2\right]$$

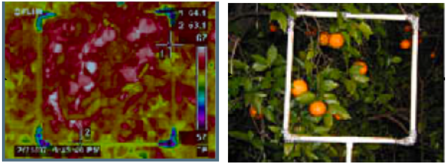
where the a priori probability $\pi(x, y|I)$ is postulated for each position (x, y) given the background information I.

The location of the object O is estimated by the mean values

$$\hat{x} = \int xp(x, y|\theta)dxdy, \quad \hat{y} = \int yp(x, y|\theta)dxdy.$$

Common representational format: Example

- automatic detection of fruits using visible images is difficult in low-light conditions etc.
- fusing visible and thermal images improves detection



(Figure: Bulanon et.al:, Biosystems Eng, 103,12-22, 2009)

- First: spatial alignment.
- Then: images are converted to common radiometric scale (8-bit gray scale)

$$I_T(i,j) = 255\frac{T(i,j) - T_{\min}}{T_{\max} - T_{\min}}$$

$$I_V(i,j) = 255\frac{R(i,j)}{R(i,j) + G(i,j) + B(i,j)}$$

Spatio-temporal transformation

- Space x and time t is mapped to common representation format as

$$(\mathrm{x}', t') = T(\mathrm{x}, t)$$

- In general,

$$T(\mathrm{x}, t) = (T_{\mathrm{x}}(\mathrm{x}, t), T_t(\mathrm{x}, t)).$$

- but often one may use the decoupled approximation

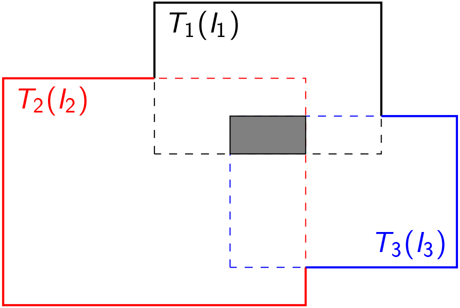
$$T(\mathrm{x}, t) = (T_{\mathrm{x}}(\mathrm{x}), T_t(t)).$$

- Sometimes this decoupling holds, i.e. when the system does not depend on time (see next example)

Spatio-temporal transformation: Example

In video surveillance and photography one needs to form a mosaic image (or panorama) of several spot images.

- spot images I_m are transformed to (object-centered) common coordinate system as $T_m(I_m)$



- mosaic $I^* = T_1(I_1) \cup T_2(I_2) \cup T_3(I_3)$ is the solid line
- spot images overlap in gray area; we stitch $T_m(I_m)$ together

Real-life example

Set of individual images:(Source: Hugin software tutorial)

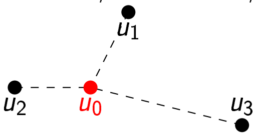
Mosaic image:

GIS: Geographical information system

- combine maps (infrastructure, demographic,...) with multiple images of the Earth from different sensors into common coordinate system
- Example: track moving objects from moving cameras
- need to describe motion of object in common coordinate system
- use absolute lat/lon coordinates, align images to that
- get absolute location of object, obtain speed of the motion

GIS: Kriging

- given sensor measurements $y_i = y(u_i), i = 1, 2, \ldots, N$ we estimate (interpolate) the value $y_0 = y(u_0)$, denoted $\hat{y}(u_0)$
- u_i is location (2D or 3D) and y is any quantity of interest: elevation, ozone level, co2 concentration,...



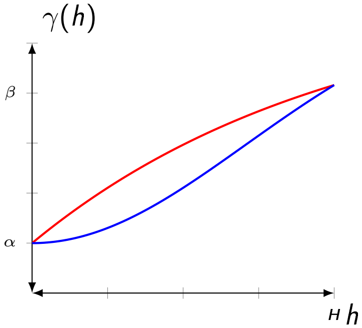
- form a linear model $\hat{y}(u_0) - \mu(u_0) = \sum_{i=1}^{N} \lambda_i (y_i - \mu(u_0))$
- minimizing the estimation variance; Kriging equations

$$\begin{pmatrix} \Sigma(u_1, u_1) & \cdots & \Sigma(u_N, u_1) & 1 \\ \vdots & & \vdots & \\ \Sigma(u_1, u_N) & \cdots & \Sigma(u_N, u_N) & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_N \\ \mu_0 \end{pmatrix} = \begin{pmatrix} \Sigma(u_0, u_1) \\ \vdots \\ \Sigma(u_0, u_N) \\ 1 \end{pmatrix}$$

- here weights sum to 1 (ordinary Kriging); $\hat{y}(u_0) = \sum_{i=1}^{N} \lambda_i y_i$

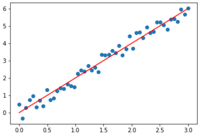
Kriging, variogram

- above the covariance function Σ is also called the variogram γ in geostatistics
- $\Sigma(u_i, u_j) = \gamma(u_i - u_j)$
- γ might take several shapes
- exponential: $\gamma(h) = \alpha - \beta(1 - \exp(-h/H))$
- Gaussian: $\gamma(h) = \alpha - \beta(1 - \exp((-h/H)^2))$
- α, β, H are called nuggett, sill, practical range

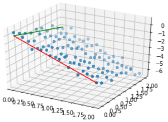


Subspace methods

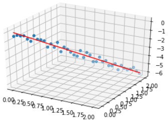
- also called dimensionality reduction
- lowers computational load and storage need, reduces overfit
- basic idea: if data lies on or near a lower dimensional linear subspace then axis of that subspace offer an effective representation of the data
- we look for directions of largest variance
- pictorially:



(a) N=2,L=1



(b) N=3,L=2



(c) N=3,L=1

UNIVERSITY OF OULU

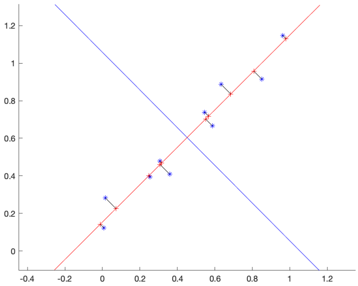
PCA: Principal component analysis

- start with input vectors $y_i, i = 1, 2, \ldots, N$ (unsupervised)
- compute mean and and sample covariance matrix

$$\mu = \frac{1}{N}\sum_{i=1}^{N} y_i, \quad \Sigma = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mu)(y_i - \mu)^T$$

- it can be shown that the L dimensional linear projection that best represents the data is $U = (u_1, u_2, \ldots, u_L)$, where u_l is an eigen vector of Σ with eigenvalue λ_l. So $\Sigma u_l = \lambda_l u_l$.
- L-dimensional representation of y_i is $\theta_i = U^T(y_i - \mu)$
- if $\lambda_1 \geq \lambda_2 \geq \cdots$ then u_1 is the first principal component, u_2 the second and so on...
- data has the most variance in the direction of u_1
- "best" means it maximizes variance or minimizes perpendicular distance to principal axis
- taking only L eigenvectors we lose some information but not too much if we take them in decreasing order of eigenvalues

PCA: Example



Red line is first principal component (line via mean of data)
(Compare with linear regression)
Blue line is the second principal component; orthogonal to first.

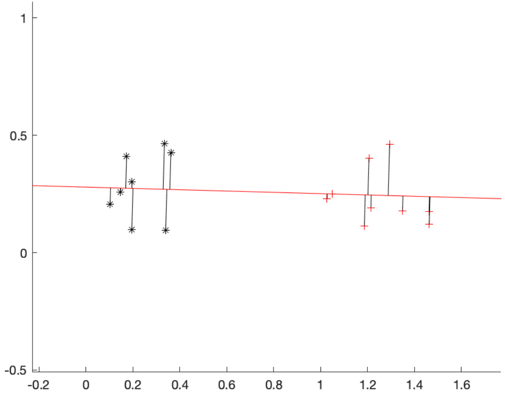
LDA: Linear discriminant analysis

- supervised technique
- input y_i is associated with given class $c_k, k = 1, 2, \ldots, K$
- denote by n_k number of input measurements in class c_k
- μ_k, Σ_k are mean vector and covariance matrix for class c_k
- now we find L-dimensional subspace U where classes are maximally separated
- u_l is eigenvector of $H = \Sigma_W^{-1}\Sigma_B$, where

$$\Sigma_B = \frac{1}{N}\sum_{k=1}^{K} n_k(\mu_k - \mu_G)(\mu_k - \mu_G)^T, \quad \Sigma_W = \frac{1}{N}\sum_{k=1}^{K} n_k\Sigma_k$$

$$\mu_G = \frac{1}{N}\sum_{k=1}^{K} n_k\mu_k$$

LDA: Example



Fusion of PCA and LDA

- test image y is to be matched against training images $Y_n, n = 1, 2, \ldots, N$
- project y, Y_n onto K-dimensional PCA and LDA subspaces:

$$\theta = (\theta(1), \ldots, \theta(K))^T, \quad \Theta_n = (\Theta_n(1), \ldots, \Theta_n(K))^T$$
$$\phi = (\phi(1), \ldots, \phi(K))^T, \quad \Phi_n = (\Phi_n(1), \ldots, \Phi_n(K))^T$$

- compute distances

$$d_n = \sum_{k=1}^{K} (\theta(k) - \Theta_n(k))^2, \quad D_n = \sum_{k=1}^{K} (\phi(k) - \Phi_n(k))^2$$

Fusion of PCA and LDA

- scale them

$$\widetilde{d}_n = \frac{d_n - \min d_n}{\max d_n - \min d_n}, \quad \widetilde{D}_n = \frac{D_n - \min D_n}{\max D_n - \min D_n}$$

- fuse by averaging

$$F_n = \frac{1}{2}(\widetilde{d}_n + \widetilde{D}_n)$$

- classify test image to class

$$n^* = \operatorname*{argmin}_n F_n$$

Multiple training sets

- used in ensemble learning
- ensemble of weak classifiers $S_m, m = 1, 2, \ldots, M$ is learnt on its own training set D_m sharing common representational format
- Bagging (bootstrap aggregating): D is bootstrapped i.e. sampled randomly with replacement using uniform probability. For example:

```
D:   1 2 3 4 5 6 7 8 9 10
D1: 5 4 10 1 4 5 7 1 2 7
D2: 3 2 1 2 6 9 8 2 1 7
D3: 5 5 4 4 10 5 8 2 1 3
```

- can be done in parallel
- learners are aggregated; for example average or majority vote
- reduces variance, eliminates overfitting

Boosting

- models are not trained independently as in bagging but iteratively (sequentially)
- each model in the sequence is trained so that more importance is given to observations misclassified by the previous model (cannot be done in parallel)
- Formally: D_{m+1} for S_{m+1} is created by resampling D such that samples that we misclassified by S_m have bigger chance of being chosen than samples that we classified correctly by S_m
- so each new model focuses on the most difficult observations
- final strong learner has lower bias