

Exercises on dimensionality reduction

Raquel Urtasun
TTI Chicago

March 15, 2010

1 Motion capture

Download the matlab toolbox of Neil Lawrence from:

<http://www.cs.man.ac.uk/~neill/mocap/>

Note that in order for this toolbox to work, you should download and add to the matlab path the NDLUTIL toolbox. See link from Neil Lawrence webpage. The motions that we are going to work today come from the CMU Graphics Lab Motion Capture Database

<http://mocap.cs.cmu.edu/>

To work with mocap, you first have to load the skeleton that describes the kinematic tree that you are going to use:

```
skel = acclaimReadSkel('examples/86.asf');
```

where the argument is the *asf* file containing the skeleton. Once the skeleton is loaded, load the motion (specification of all the rotations and global motion) by using the following command:

```
[channels, skel] = acclaimLoadChannels('examples/86_10.amc', skel);
```

where the first argument is the location of the *amc* file that contains the motion, and the second argument is the skeleton that specifies the kinematic tree. To play the animation use the following command

```
skelPlayData(skel, channels, 1/120);
```

where you specify the kinematic tree, the motion and the framerate to play. The mocap was capture at 120 frames per second.

Download 3 walking motions from different subjects. Segment the walking motions to have exactly one cycle per motion. We will use a single skeleton to play all the motions. This might result in motions that penetrate the ground plane. We will ignore retargeting (adapt the joint angles to a new skeleton) in the exercises. Downsample the mocap from the original 120Hz to 30Hz. How should you do this? What are the problems that might occur if you do this?

2 Linear models

Create a function that using the matlab *svd* command computes PCA. The entries to this function are $\mathbf{Y} \in \mathbb{R}^{N \times D}$, the matrix computed as the concatenation of the N training examples, with $\mathbf{Y} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T$, and the number of desired components d

$$[\mathbf{U}, \boldsymbol{\Sigma}] = \text{computePCA}(\mathbf{Y}, q)$$

The function returns the basis \mathbf{U} , and a vector $\boldsymbol{\Sigma}$ composed of the singular values. What's the relationship between $SVD(\mathbf{X})$ and $computePCA(\mathbf{X})$? How could you compute PCA efficiently when the number of dimensions is much smaller than the number of examples? Modify your function *computePCA* so that it is efficiently implemented.

Apply PCA to the database of walking motions that you have previously download, i.e., one cycle of walking of 3 different subjects. Can you interpret the results? Remove the global motion, what's the result now? Learn 2D and 3D latent spaces. How can you determine the dimensionality of the latent space?

3 Graph-based methods

In this section we will experiment with graph-based methods. Choose two techniques from the following table and download them

| method | URL |
|---------------------|---|
| Isomap | http://isomap.stanford.edu/ |
| LLE | http://ttic.uchicago.edu/~rurtasun/courses/ETH10/lle.m |
| Laplacian Eigenmaps | http://ttic.uchicago.edu/~rurtasun/courses/ETH10/laplacian.tar |
| MVU | http://www.cse.wustl.edu/%7Ekilian/Downloads/MVU.html |

For the installation of MVU, follow the instructions on the webpage for installing SDP solver. Use the following wrappers develop by Carl Ek to have a unified data structure:

$$\text{http://ttic.uchicago.edu/~rurtasun/courses/ETH10/carl.tar}$$

Use the wrapper functions for the 2 algorithms of your choice as well as the *MDS* wrapper to compute low dimensional representations of the mocap walking dataset. Do it by first using a single example (1 cycle of 1 subject). What do you see? how do you interpret this? Now learn the latent space by using the full dataset (1 cycle of 3 subjects). What do you see now? How do you interpret this? Repeat the experiments (with 1 subject and 3 subjects) by removing the global motion. What do you see now?

4 Non-linear latent variable models

Download the Gaussian process latent variable model (GPLVM) from Neil Lawrence website

$$\text{http://www.cs.man.ac.uk/~neill/}$$

Follow the instructions to install depending packages. Add the corresponding paths to matlab by using *addpath* function. Use Carl Ek's wrapper

$$[ll, model] = \text{gplvm_modelscore}(\mathbf{Y}, \mathbf{X}, \text{nr_iters})$$

to learn GPLVMs. The input for this function are the data \mathbf{Y} , and initialization to the latent space \mathbf{X} and the number of iterations to perform. The outputs are the log likelihood l , and the learned model. The latent space learned is stored in *model.X*. Learn GPLVMs with initialization the different latent spaces that you learned either using PCA or graph-based methods. What do you observe? what can you conclude?