Exercise Sheet #10

Due: 2020-01-16, 16:00

Fred Hamprecht
Manuel Haußmann

General Regulations.

- You should hand in your solutions in groups of at least two people (recommended are three).
- The theoretical exercises can be either handwritten notes (scanned), or typeset using LaTeX.
- Practical exercises should be implemented in python and submitted as jupyter notebooks (`.ipynb`). Always provide the (commented) code as well as the output, and don't forget to explain/interpret the latter.
- Submit all your files in a single `.zip` archive to `mlhd1920@gmail.com` using the following standardized format: The subject line should consist of the full names of all team members as well as the exercise, and the title of the zip archive the last names. I.e. assuming your group consists of Ada Lovelace, Geoffrey Hinton and Michael Jordan, this means

  Subject: [EX10] Michael Jordan, Geoffrey Hinton, Ada Lovelace
  Zip Archive: `ex10-jordan-hinton-lovelace.zip`

1 Kalman Filter (10 pt)

In the lecture we discussed the following linear dynamical system, consisting of latent variables $\mathbf{z}$ and observable variables $\mathbf{x}$, following the relationship

$$\mathbf{z}_1 = \boldsymbol{\mu}_0 + \mathbf{u}, \qquad \text{where} \quad \mathbf{u} \sim \mathcal{N}(\mathbf{u}|0, \mathbf{P}_0) \tag{1}$$

$$\mathbf{z}_t = \mathbf{A}\mathbf{z}_{t-1} + \mathbf{w}_t, \qquad \text{where} \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{w}_t|0, \boldsymbol{\Gamma}) \tag{2}$$

$$\mathbf{x}_t = \mathbf{C}\mathbf{z}_t + \mathbf{v}_t, \qquad \text{where} \quad \mathbf{v}_t \sim \mathcal{N}(\mathbf{v}_t|0, \boldsymbol{\Sigma}). \tag{3}$$

We assume $\boldsymbol{\mu}_0, \mathbf{P}_0, \boldsymbol{\Gamma}, \boldsymbol{\Sigma}, \mathbf{A}$, and $\mathbf{C}$ to be known.[1] Slightly reformulating the results from the lecture we get that $p(\mathbf{z}_t|\mathbf{x}_{1:t}) \sim \mathcal{N}(\mathbf{z}_t|\boldsymbol{\mu}_t, \mathbf{V}_t)$ with

$$\boldsymbol{\mu}_t = \mathbf{A}\boldsymbol{\mu}_{t-1} + \mathbf{K}_t(\mathbf{x}_t - \mathbf{C}\mathbf{A}\boldsymbol{\mu}_{t-1}) \tag{4}$$

$$\mathbf{V}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{C})\mathbf{P}_{t-1}, \tag{5}$$

where $\mathbf{I}$ is the identity matrix of suitable dimensionality. $\mathbf{P}_{t-1}$ and $\mathbf{K}_t$ are given as

$$\mathbf{P}_{t-1} = \mathbf{A}\mathbf{V}_{t-1}\mathbf{A}^T + \boldsymbol{\Gamma} \tag{6}$$

$$\mathbf{K}_t = \mathbf{P}_{t-1}\mathbf{C}^T(\mathbf{C}\mathbf{P}_{t-1}\mathbf{C}^T + \boldsymbol{\Sigma})^{-1}. \tag{7}$$

i) Consider the situation where the observation noise goes to zero for $\mathbf{x}_t$, i.e. we now have a time dependent $\boldsymbol{\Sigma}_t$, with $\boldsymbol{\Sigma}_j = \boldsymbol{\Sigma}$ for $j = 1, ..., t-1$ and $\boldsymbol{\Sigma}_t = \mathbf{0}$. Assuming $\mathbf{C} = \mathbf{I}$, what do we get for $p(\mathbf{z}_t|\mathbf{x}_1, ..., \mathbf{x}_t)$? Does that agree with your intuition?

ii) Implement the Kalman filter updates and plot both the observations $\mathbf{x}$ as well as the estimated means $\boldsymbol{\mu}_t$ you get for `kalman.npy`. We assume the following system. $\mathbf{z} \in \mathbb{R}^4$ with two spatial dimensions and

[1] We haven't discussed the case where we want to learn those in the lecture. If you want to know how, have a look e.g. at *Bishop* 13.3.2.

the other two giving the velocities in each of the directions. We assume

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \tag{8}$$

i.e. we observe only the locations. The initial location and velocity we assume to be given as $\boldsymbol{\mu}_0 = [8, 10, 1, 0]$ and the noise covariances are given as $\mathbf{P}_0 = \mathbf{I}_4$, $\Gamma = 0.001 \cdot \mathbf{I}_4$, and $\mathbf{\Sigma} = \mathbf{I}_2$, where the subscript indicates the dimensionality of the corresponding identity matrix.

2 Ilastik + Multicut for Instance Segmentation (20 pt)

In the lecture, we discussed how we get from raw images to segmented cells. The goal of this exercise is to walk through the whole pipeline from raw images to the final segmentation using the ilastik software.[2]

i) Download and setup Ilastik, as well as the *Helmstaedter* data set, a 3d data set of the mouse retina.[3]

ii) In order to segment the individual cells in the second part of the pipeline we first need a good boundary predictor. Following along with the *Pixel Classification* tutorial on the website,[4] interactively label some cells and cell boundaries to train such a classifier.

iii) Finally, use the multicut workflow,[5] to end up with a final cell segmentation.

For each step document your results via screenshots.

3 Integer Linear Program (10 pt)

The image partition problem as discussed in the lecture comes down to

$$\arg\min_{\mathbf{z}} \underbrace{\mathbf{w}^T\mathbf{z}}_{=:\mathcal{L}(\mathbf{z})} \quad \text{s.t.} \quad \mathbf{z} \text{ gives a valid clustering and } \mathbf{z} \in \{0,1\}^{|E|}. \tag{9}$$

Assuming we are given the graph in Figure 1, with weights $\mathbf{w} = (w_{ab}, w_{ac}, w_{bd}, w_{cd})$ and edge cut indicators $\mathbf{z} = (z_{ab}, z_{ac}, z_{bd}, z_{cd})$, give the loss $\mathcal{L}(\mathbf{z})$ for all combinations of $\mathbf{z}$ and the final result subject to the validity constraints. The weight vector is given as $\mathbf{w} = (0.1, 0.6, 0.9, -1.2)$.

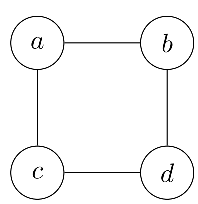

Figure 1: The graph.

[2] www.ilastik.org
[3] See https://www.ilastik.org/download.html for both.
[4] https://www.ilastik.org/documentation/pixelclassification/pixelclassification
[5] Follow this tutorial https://www.ilastik.org/documentation/multicut/multicut