DD2434 Machine Learning, Advanced Course
Assignment 3: Sampled and Ensemble Models

Hedvig Kjellström

Deadline December 17, 2014

The assignment will be reviewed by an oral examination on the 18th and the 19th of December. There will be time-slots to sign up for on the course web page in time for the review. You should submit your report on the 17th of December before 12 : 00 NOON CET so that I have a chance to look through what you have done before the oral review. In the beginning of the review session, the examiner will ask you what grade you aim for, and ask questions related to that grade. All the tasks have to be presented at the same review session you can not complete the assignment with additional tasks after it has been examined and given a grade. Come prepared to the review session! The review will take 15 minutes or less, so have all your results in order. The grading of the assignment will be as follows,

E Completed Task 3.1 and 3.2.

D E + Completed Task 3.3.

C D + Completed Task 3.4.

B C + Completed Task 3.5.

A B + Completed Task 3.6.

These grades are valid for review December 17, 2014. See the course web page, HT 2014 - Assignments in the menu, for grading of delayed assignments.

Instructions for Lab3

You are supposed to write down the answers to the specific questions detailed for each task in a report. This report should clearly show how you have implemented the methods, how you have conducted experiments and drawn conclusions from those - it should be possible to reproduce your graphs based on the explanation in the report. The report should be self-explanatory – that is, we should be able to understand what you did, just from reading the report. Your assumptions if any should be stated clearly. For the practical part of the task you should not show any of your code but rather only show the results of your experiments by using images and graphs. It should be evident from the report that you did all the required implementations, and the explanation of results in the report should be self-explanatory – that is, you should not say that further information will be given in the oral examination.

Abstract

In this assignment you will be acquainted with sampling and ensemble based methods for classification and estimation. Common for these types of methods is that distributions and functions are represented, either as a collection of samples, or as an ensemble of local simple functions, that together are able to represent the characteristics of the entire global distribution or function. This is a conceptually different approach from the one studied in Assignment 2, where you represented distributions or functions analytically.

More specifically, you will implement kNN and AdaBoost classification, and compare these. You will also implement a particle filter, and experiment with different motion priors.

Furthermore, you will learn about how to conduct experiments, evaluating the performance of methods as a function of parameters in the methods in a systematic manner – a very important aspect of technical sciences – and learn about performance measures such as confusion matrices and ROC curves.

Datasets

2D Wave Dataset. A toy dataset has been developed for Tasks 3.1-3.4, consisting of a binary distribution $y \in [0, 1]$ over a continuous 2D state-space $\mathbf{x}$ (Figures 1 and 2). The boundary between the (red) $y = 0$ are and the (green) $y = 1$ area is somewhat non-linear. The analytic expression of the boundary is $\mathbf{b} = r[\cos(\text{abs}(\alpha)), \sin(\text{abs}(\alpha))]$ where the radius $r \in [-1, 1]$ and the angle $\alpha = \pi r$ (rad). This is to be used in your plots as a ground truth baseline for the classifier boundaries found by your methods.

A sample from this distribution can be expressed as a tuple $(\mathbf{x}_i, y_i)$ where $\mathbf{x}_i$ is a position in the 2D statespace and y_i is the binary label of this sample. By drawing such samples from this distribution, we have created a collection of datasets $D_{N,\sigma}$, parameterized by N, the number of samples, and σ, the standard deviation of a random Gaussian noise term $\nu(\sigma)$ that has been added to $\mathbf{x}$. (This simulates measurement error, e.g., due to sensor noise.) Thus, $D_{N,\sigma} = \{(\mathbf{x}_i + \nu(\sigma), y_i)\}_{i=1}^{N}$, a set of cardinality N, consisting of points $(\mathbf{x}_i + \nu(\sigma), y_i)$.

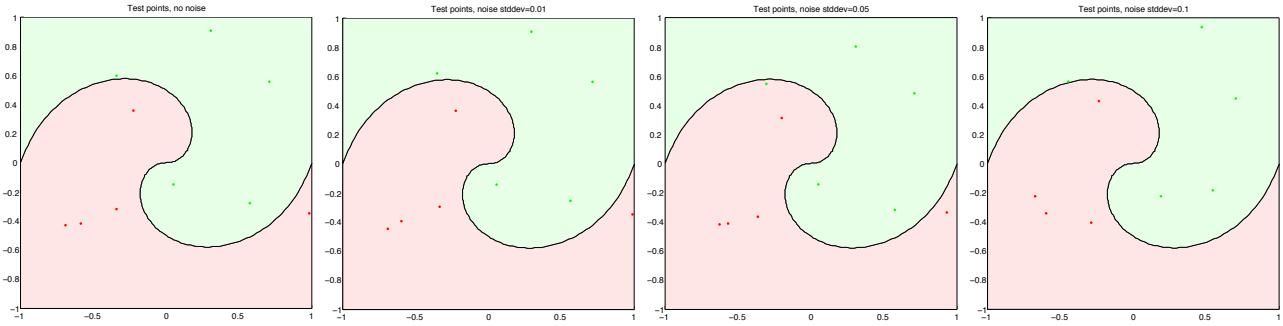
Training and test sets were generated, and can be retrieved from the text file `wave.txt`, found in the Assignments page in the course web page. Figure 1 shows the test sets, while Figure 2 shows the different training sets.

2D Swiss Roll Dataset. We also have a more non-linear dataset for Tasks 3.1-3.4, consisting of a binary distribution $y \in [0, 1]$ over a continuous 2D state-space $\mathbf{x}$ (Figures 3 and 4). The boundary between the (red) $y = 0$ are and the (green) $y = 1$ area is here highly non-linear. The analytic expression of the boundary is $\mathbf{b} = r[\cos(\text{abs}(\alpha)), \sin(\text{abs}(\alpha))]$ where the radius $r \in [-1, 1]$ and the angle $\alpha = 3\pi r$ (rad). This is to be used in your plots as a ground truth baseline for the classifier boundaries found by your methods.

A sample from this distribution can be expressed as a tuple $(\mathbf{x}_i, y_i)$ where $\mathbf{x}_i$ is a position in the 2D statespace and y_i is the binary label of this sample. By drawing such samples from this distribution, we have created a collection of datasets $D_{N,\sigma}$, parameterized by N, the number of samples, and σ, the standard deviation of a random Gaussian noise term $\nu(\sigma)$ that has been added to $\mathbf{x}$. (This simulates measurement error, e.g., due to sensor noise.) Thus, $D_{N,\sigma} = \{(\mathbf{x}_i + \nu(\sigma), y_i)\}_{i=1}^{N}$, a set of cardinality N, consisting of points $(\mathbf{x}_i + \nu(\sigma), y_i)$.

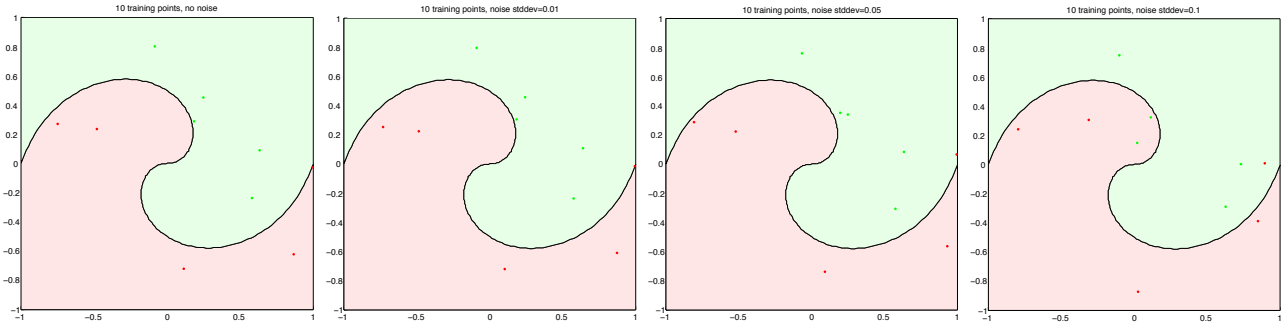
Training and test sets were generated, and can be retrieved from the text file `swissRoll.txt`, found in the Assignments page in the course web page. Figure 3 shows the test sets, while Figure 4 shows the different training sets.

2D City With Car Dataset. For Tasks 3.5-3.6 we have constructed a dataset depicting a car driving in a city with rectangular blocks (Figure 5). The dataset consists of a ground truth 2D car trajectory $\mathbf{x}_{0:T} = [\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_T]$ (depicted by red trajectories in Figure 5). For each time step $t \in [1, T]$, there is an observation $\mathbf{y}_t = \mathbf{x}_t + \nu(\sigma)$, i.e., a noisy version of the true state $\mathbf{x}_t$ with a random Gaussian noise term $\nu(\sigma)$ with standard deviation σ (depicted by black dots in Figure 5(a-e)). In our dataset, stored in the text file `cityWithCar.txt`, found in the Assignments page in the course web page, $T = 100$ and $\sigma = 1$.

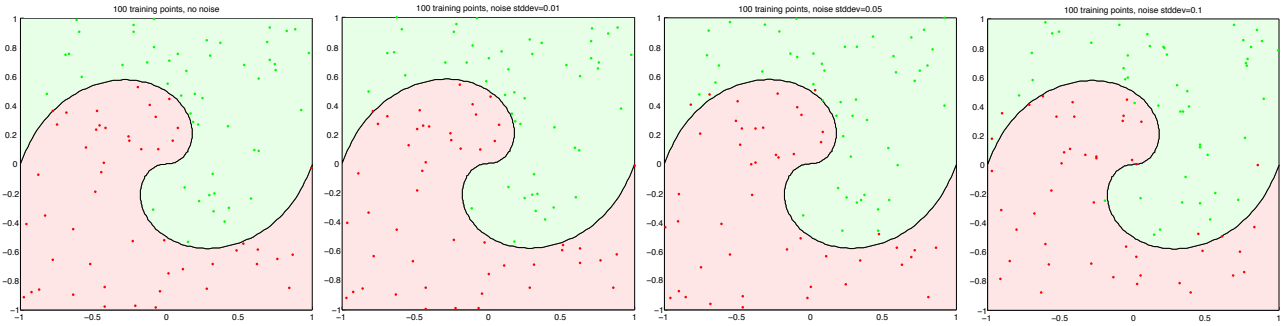


(a) $\sigma = 0$ (b) $\sigma = 0.01$ (c) $\sigma = 0.05$ (d) $\sigma = 0.1$

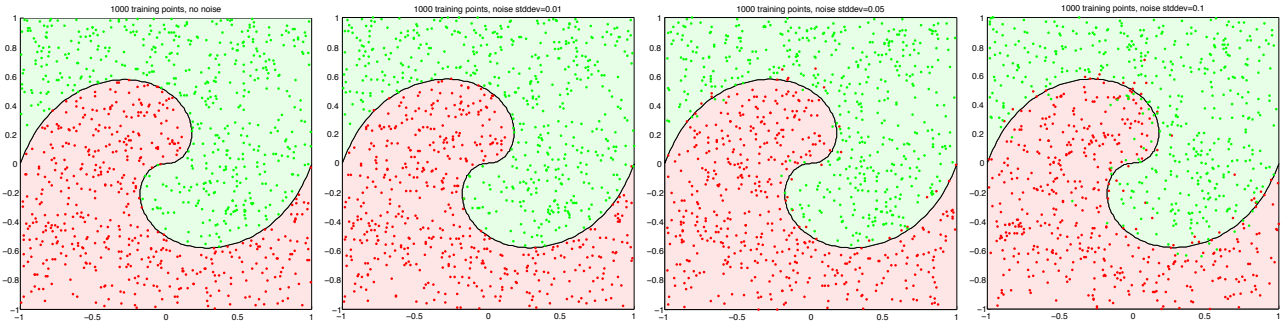
Figure 1: The different Wave test sets $D_{\sigma}^{\mathbf{test}}$. $N = 10$. Red denotes $y = 0$, while green denotes $y = 1$.



(a) $N = 10$, $\sigma = 0$ (b) $N = 10$, $\sigma = 0.01$ (c) $N = 10$, $\sigma = 0.05$ (d) $N = 10$, $\sigma = 0.1$

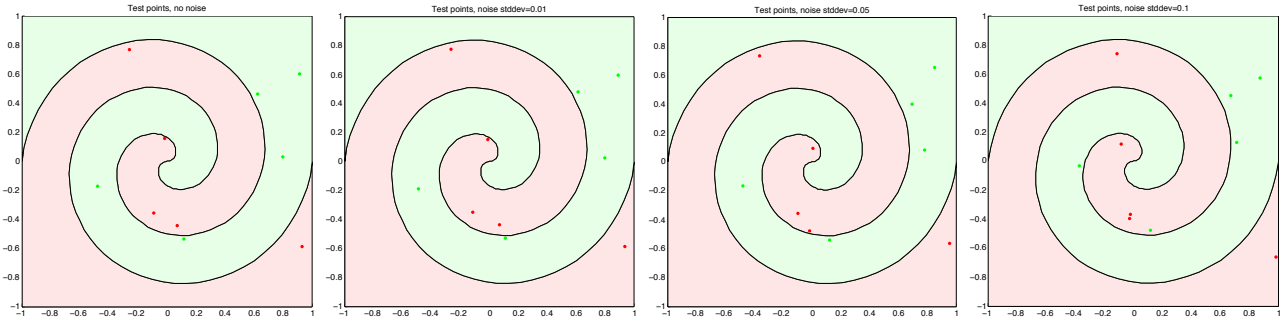


(e) $N = 100$, $\sigma = 0$ (f) $N = 100$, $\sigma = 0.01$ (g) $N = 100$, $\sigma = 0.05$ (h) $N = 100$, $\sigma = 0.1$



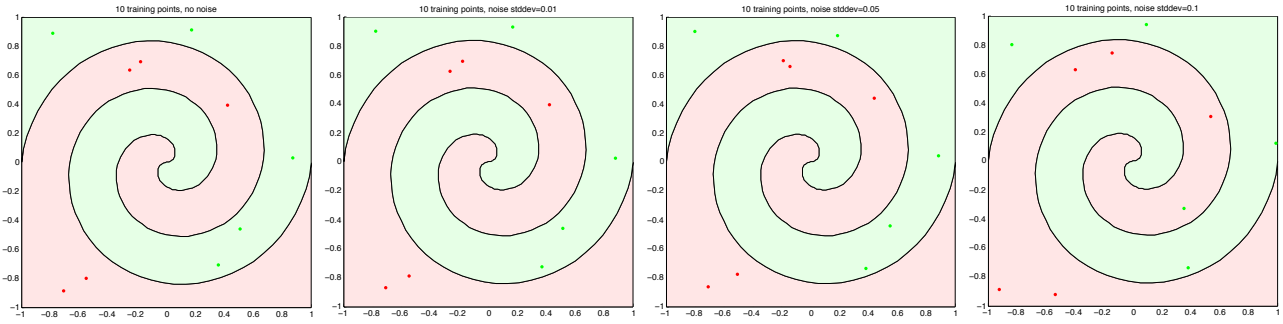
(i) $N = 1000$, $\sigma = 0$ (j) $N = 1000$, $\sigma = 0.01$ (k) $N = 1000$, $\sigma = 0.05$ (l) $N = 1000$, $\sigma = 0.1$

Figure 2: The different Wave training sets $D_{N,\sigma}$. Red denotes $y = 0$, while green denotes $y = 1$.

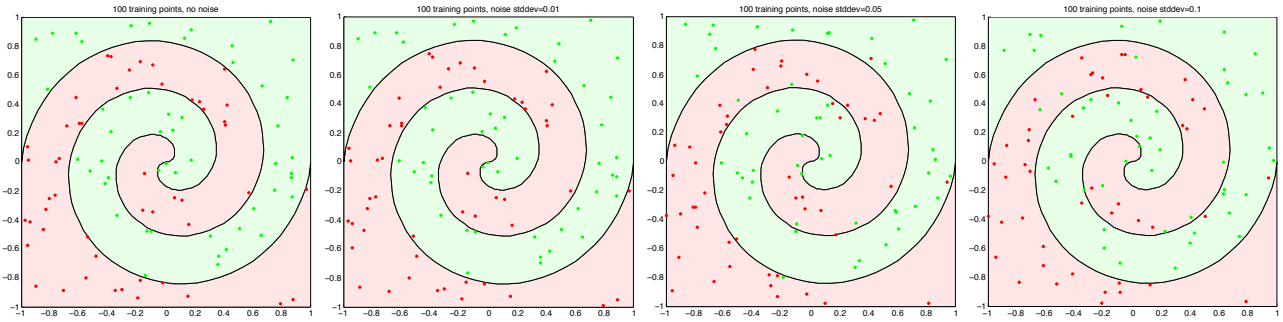


(a) $\sigma = 0$ (b) $\sigma = 0.01$ (c) $\sigma = 0.05$ (d) $\sigma = 0.1$

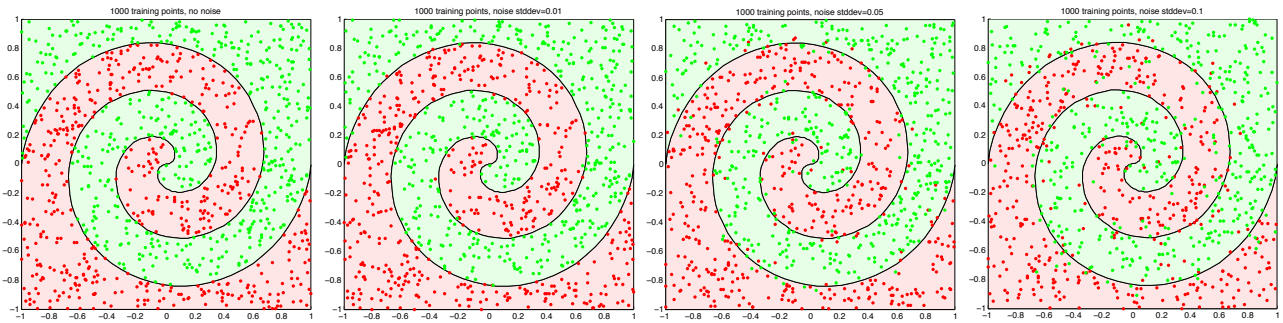
Figure 3: The different Swiss Roll test sets $D_\sigma^{\mathbf{test}}$. $N = 10$. Red denotes $y = 0$, while green denotes $y = 1$.



(a) $N = 10$, $\sigma = 0$ (b) $N = 10$, $\sigma = 0.01$ (c) $N = 10$, $\sigma = 0.05$ (d) $N = 10$, $\sigma = 0.1$

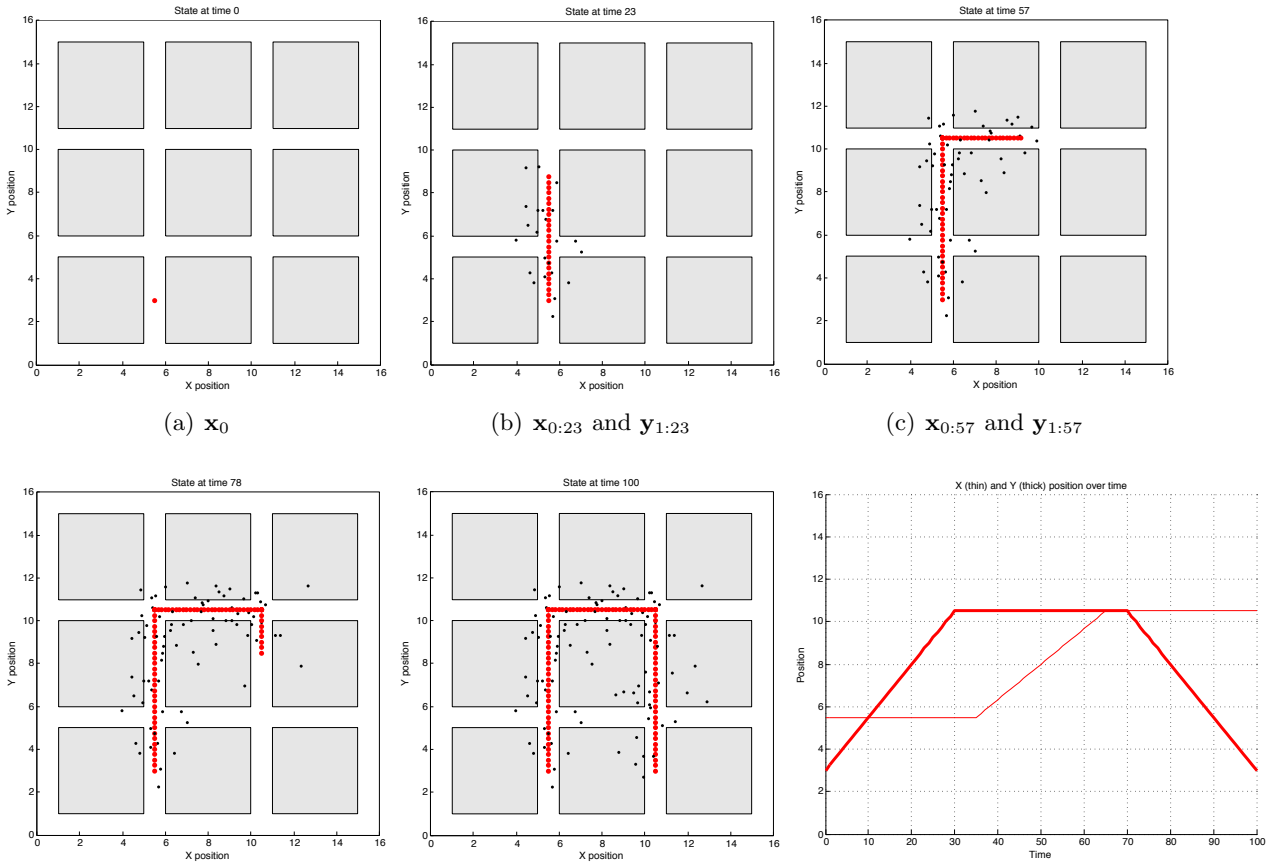


(e) $N = 100$, $\sigma = 0$ (f) $N = 100$, $\sigma = 0.01$ (g) $N = 100$, $\sigma = 0.05$ (h) $N = 100$, $\sigma = 0.1$



(i) $N = 1000$, $\sigma = 0$ (j) $N = 1000$, $\sigma = 0.01$ (k) $N = 1000$, $\sigma = 0.05$ (l) $N = 1000$, $\sigma = 0.1$

Figure 4: The different Swiss Roll training sets $D_{N,\sigma}$. Red denotes $y = 0$, while green denotes $y = 1$.



(a) $\mathbf{x}_0$ (b) $\mathbf{x}_{0:23}$ and $\mathbf{y}_{1:23}$ (c) $\mathbf{x}_{0:57}$ and $\mathbf{y}_{1:57}$

(d) $\mathbf{x}_{0:78}$ and $\mathbf{y}_{1:78}$ (e) $\mathbf{x}_{0:100}$ and $\mathbf{y}_{1:100}$ (f) $x^{\mathrm{X}}_{0:T}$ and $x\mathrm{Y}_{0:T}$

Figure 5: The dataset $(\mathbf{x}_{0:T}, \mathbf{y}_{1:T})$. (a-e) The city landscape with the ground truth trajectory $\mathbf{x}_{0:T}$ in red and the observations $\mathbf{y}_{1:T}$ in black, at different time steps in the sequence. (b) The values of the ground truth $x^{\mathrm{X}}_{0:T}$ and $x\mathrm{Y}_{0:T}$, $\mathbf{x}_t = [x^{\mathrm{X}}_t, x^{\mathrm{Y}}_t]$.

Figure 5 depicts the dataset $(\mathbf{x}_{0:T}, \mathbf{y}_{1:T})$ and the city environment with white streets and grey blocks. For tracking applications, the initial position $\mathbf{x}_0$ can be considered known, while the following positions $\mathbf{x}_t$ are hidden and only observed through the noisy measurements $\mathbf{y}_t$.

3.1 kNN Classification, Implementation

The first task of this assignment is to implement a k nearest neighbor (kNN) classifier, and employ it on the Wave, and if you like, on the Swiss Roll dataset.

We will start with the special case $k = 1$. Implement a function that, given a training dataset $D_{N,\sigma}$, outputs a Voronoi diagram of the corresponding 1NN classifier, with cells with value $y = 0$ painted red and cells with value $y = 1$ painted green. Compare to the ground truth boundary. (TIP: You can plot the ground truth boundary using the analytic expression above, and thereby recreate the plots in Figures 1-4).

Question 1: *Plot the Voronoi diagrams of all training datasets $D_{N,\sigma}$. Compare these to the ground truth boundary. How do you think the classification rate (i.e., the ratio between the number of correctly classified test samples and the total number of test samples) will vary with N and σ?*

Now make your classifier more general, so that it handles classification with $k > 1$. The kNN classifier will be systematically evaluated in the nest task.

3.2 kNN Classification, Experiments

You will now get (re)acquainted with systematic experimental evaluation.

Run your classifier developed in Task 3.1 on all training datasets $D_{N,\sigma}$ of the Wave, and if you like, on the Swiss Roll dataset. For each dataset, run the classifier with a range of k values. (TIP: Always use the same value of σ for the test and training data. For each test dataset $D_{\sigma}^{\mathbf{test}}$, you can thus perform experiments with 3 different training datasets $D_{N,\sigma}$.)

For each trial case $D_{N,\sigma}$, compute the confusion matrix and the classification rate - these are measurements of the classifier performance. Make plots over classification rate as a function of the three parameters.

Question 2: *Make observations: How does the classification rate vary with the three parameters k, N, and σ? Are there correlations between parameters? What are the optimal values?*

Question 3: *It is obviously better with more data, i.e., higher N. However, discuss the benefits of a lower N.*

Question 4: *The classification will obviously be more robust for higher k since the voting becomes more informed. Discuss the benefits of a lower k, in relation to N.*

3.3 AdaBoost Classification, Implementation (D-A)

In this task, you will implement an ensemble based classification method, AdaBoost, and employ it on the Wave, and if you like, on the Swiss Roll dataset. (Note that it will not find the boundary completely on the Swiss Roll dataset, so do not use this for debugging.)

Let the weak learners be linear classifiers ϕ that put out boundaries of any orientation and position in the 2D state space (i.e., linear classifiers with two parameters). Let M be the number of steps in the cascade, ϕ_m be the classifier at step m, and α_m the corresponding weight.

Question 5: *Train your AdaBoost classifier with each training dataset $D_{N,\sigma}$ (not all of them together, but a separate experiment for each N, σ). For each step m in the cascade, plot the linear decision boundary ϕ_m with the dataset. Compare the collection of boundaries to the ground truth boundary. How do you think the classification rate (i.e., the ratio between the number of correctly classified test samples and the total number of test samples) will vary with M, N, and σ?*

3.4 AdaBoost Classification, Experiments (C-A)

Perform the same type of experiments as in Task 3.2, investigating the effect of M, N, and σ on the classification rate.

Make observations: How does the classification rate vary with the three parameters M, N, and σ? Are there correlations between parameters? What are the optimal values?

Question 6: *Compare the two classifiers. What are the problems with each one? When would you prefer one or the other?*

Question 7: *Why is very large N (in the order of 1M and up) less of an issue with AdaBoost than with kNN?*

3.5 Particle Filter Estimation, Linear Constant Velocity Motion Model (B-A)

We now change focus and study another setting, sequential estimation. In this task and the next you will implement a particle filter, and perform experiments with the City With Car dataset. Less guidance is given in the last two tasks, since we expect that students aiming for the grades of B or A should show ability to structure problems and search for information autonomously.

The algorithm you will implement is the most basic particle filtering algorithm, where

- the initial position is considered known, i.e., the estimated initial position $\hat{\mathbf{x}}_0 \equiv \mathbf{x}_0$,
- the proposal distribution is sampled only from the prior,
- the particle set is resampled in every time step,
- a little Gaussian noise is added to each particle after resampling, to mitigate sample impoverishment,
- the motion model is that of an object moving on a frictionless surface and being pushed a little now and then: The most probable motion is to maintain constant velocity, and the probability of deviations from the constant velocity (i.e., accelerations) are normally distributed around a mean of acceleration 0.

(TIP: you should use the same observation model as was used to generate the data – this will statistically produce the best tracking performance. However, the motion model with which the data was actually generated is not known.)

(TIP: Think about what variables you need in your statespace for the constant velocity motion model to work. It will not be enough with just the 2D position x_t.)

Question 8: *Describe one time step of your particle filter, using proper notation and all the necessary equations to make the algorithm re-implementable just from reading the report. Run the algorithm on the City With Car dataset, for a* **number of different parameter settings in the motion model***, and visualize the results by 1) plotting the estimated tracks $\hat{\mathbf{x}}_{0:T}$, 2) by plotting the particle clouds for every 3rd time step in each sequence (or alternatively, generating a movie of all time steps in each sequence).*

Question 9: *Define a goodness measure for the tracking, and present the goodness of the performance with different parameter settings in the motion model.*

Question 10: *In a more qualitative sense, what problems can you observe with different parameter settings in the motion model?*

3.6 Particle Filter Estimation, More Informative Motion Model (A)

In the previous task you noted some flaws in the tracking, that had to do with the motion model – more specifically, with the fact that the constant velocity motion model in the tracker was a gross simplification of the true motion model of a car moving in a city.

The task here is now to define a motion model that takes into account everything you know about cars and driving in cities.

Question 11: *Make a bullet list of things you know about how cars move in cities (Where do they drive? Where do they speed up, slow down, stop?)*

Question 12: *Describe, using the necessary mathematical notation, the best motion model for the city in the City With Car dataset.*

Question 13: *Redo all experiments from Task 3.5, but now with the new, more informative motion model. How does the performance improve – both quantitatively and qualitatively?*

Good Luck!