1 Goals

In this practical, you will study another application of some algorithms you have seen to perform non-linear regression. While in a classification task the goal is to tell whether a sample belongs to a specific class, regression algorithms try to learn a (usually) continuous function. Once the algorithm has been trained, it is possible to estimate the value of the learned function for each sample in the input space. Here we will cover 2 machine learning techniques that perform regression: Gaussian Mixture Regression (GMR) and Support Vector Regression (SVR).

For this practical, you must download another version of MLDemos from the link http://lasa.epfl.ch/teaching/lectures/ML_Msc/MLDemos_Master.zip

This assignment will be graded through an oral presentation that will take place on December 12. This presentation counts for 10% of the total grade of the course. The practical session are to be done in teams of two to three people. Unless told otherwise, we assume that the work has been shared equally by the members of the team and hence all members will be given the same grade.

Instructions regarding the format of the oral presentation have been sent to you by email and are repeated at the bottom of this assignment sheet.

2 Gaussian Mixture Regression

Gaussian Mixture Regression (GMR) proceeds in two steps. First, it trains a Gaussian Mixture Model (GMM) which models the joint probability density of both input and output data, i.e. \( p(x, y) \). The regressive signal at testing is computed by taking the expectation of the GMM conditioned on the known value of the input dimensions. E.g., if we query for \( y \), given the input \( x \) we compute \( E[p(y|x)] \).

\[
p(y) \quad \text{GMM (joint distribution)} \\
p(y|x) \quad \text{GMM (conditional distribution)} \\
\hat{y} = E[p(y|x)] \quad \text{GMR (function)}
\]

In addition to the regressive signal of GMR, one can also compute the variance associated to each regressive output, i.e. \( \text{var}(p(y|x)) \). Figure 1 shows an example of the GMR output given
by MLDemos. The central line is the regressive signal, whereas ±1 and 2 std of the variance are highlighted with the lighter lines surrounding the regressive signal. The shading corresponds to the likelihood of the associated pdf. Recall that the variance of the regressive model \( \text{var}\{p(y|x)\} \) gives information on the relative dispersion of the predictive model. The confidence or uncertainty of the model is, however, given by the likelihood. Far from the training data, the confidence of the model will be low. However, the variance associated to the regressive signal can be small.

Beware that, while, in principle, GMR can regress on a multi-dimensional output, in MLDemos, you can choose only one dimension for the regressive signal.

Figure 1: Regression using GMR. The grayscale background shows the likelihood in the different zones of the input/output domain. This allows to estimate how confident the output of the regression is.

3 Support Vector Regression

As in Support Vector Machine for classification, Support Vector Regression (SVR) chooses a subset of the datapoints as support vectors. These points are used in combination to determine the regressive signal, i.e. \( y = f(x) = \sum_i \alpha_i k(x^i, x) \), with \( \alpha_i \) positive scalars, and \( k(x^i, x) \) the kernel function computed between the query point \( x \) and each training point \( x^i \). To account for the noise inherent to the data, SVR introduces a parameter, \( \epsilon \), that determines the uncertainty in the prediction. The predictive output \( y \) can be estimated up to an error of \( \pm \epsilon \). complexity of the function (see Figure 2). The \( \epsilon \) can be estimated automatically when using \( \nu \)-SVR.

Figure 2: Regression using SVR. The image shows different values for the \( \epsilon \) parameter, a tube is drawn around the mean corresponding to \( f(x) \pm \epsilon \). A large tube does not need to be very complex in shape to contain most samples, whereas a smaller one will result in a more complex shape.

4 Getting started

Each group will be assigned a dataset to work on. The datasets can be downloaded from the link http://lasa.epfl.ch/teaching/lectures/ML_Msc/Datasets.zip. The description of each data...
set and their assignment to groups can be found in the *datasets.txt* file.

I) **Draw/Load your data:** For the two parts of this assignment, you will need to either draw data on the screen or load the “.ml” file assigned to your group using File→Load.

II) **Choose the input dimensions and the output dimension:** From the Algorithm or Compare window, click on the *Input Dimensions* button and select every dimension that you would like to use as an input of your regression algorithm (these dimensions are detailed in the *Datasets.txt* file for your assigned dataset). Also, choose the output dimension by selecting the *Regression Dimension* (see Figure 3). *Manual Selection* enables you to remove specific datapoints but it is not used here.

![Figure 3: Select the Input and output dimensions](image)

III) **Run the regression algorithm and compare the algorithms** This works the same way as for classification. In the comparison window, you will compare the *error* for each algorithm (MSE), and the error’s variance.

5 **What to do**

For this task, you will use 2 datasets, one for a *qualitative* assessment of the algorithms, one for the *quantitative* evaluation of their performance. First, you will manually draw a 2D dataset to study *graphically* the effect of changing the parameters of the two algorithms (see Figures 1 and 2). Try to find a function that shows explicitly the advantages or disadvantages of each method. Use this 2D dataset to make a *qualitative* evaluation of the algorithms.

Second, you will be assigned a regression dataset on which you will perform the *quantitative* evaluation. The regression error is computed in terms of Mean Square Error (MSE) between the testing sample real value and its estimation. Moreover, the variance of the errors is provided. This is done to get a feeling of how well the algorithm is modelling the underlying function. Indeed a method with a small error but a large variance may be less preferable than a method with a larger error but a very small variance: the second one is not good but at least it is reliable!

For both datasets, study the performance using different parameters:

- **GMR:** test the 3 types of covariance matrices; change the amount of components per class and the initialization method.
- **ε-SVR**: test the 3 types of SVM kernels; change the kernel degree or width (depending on the kernel); change the penalty factor $C$ (testing 1-100); change the size of the $\epsilon$-tube.

- **Train/Test ratio**: test the algorithms using different ratios of training/testing samples.

Run cross-validation on these parameters using the “Compare” window in MLDemos to obtain an estimation of the variance of the results you obtain. Generate graphs and tables of the results you obtain.

### 6 Presentation

The oral presentation of this practical should last no more than 15 minutes for groups of 3 students (10 minutes for group of 2 students). You should share the presentation so that each of you present for 5 minutes to ensure an even split. We will clock you and stop you if you exceed the allocated time even if you are not done with the presentation. The presentation will be followed by 5 minutes of questions.

Presentations should not explain what each technique does but should report solely on what you have done. It should include a short introduction to explain what the datasets you use consist of (number of datapoints, dimensions, particularity of dataset if relevant). Then you should explain what you tested (choice of variable, comparison across techniques) and what results you obtained. Whenever possible, tell us what your expectation were regarding performance of one particular technique for one particular dataset and then discuss whether these expectations matched the results.

If you have two members in the team, plan a 10 minutes presentation. A good rule of thumb is little less than 1 slide per minute, so you should limit yourself to 8-9 slides. A possible structure for a presentation might be:

- introduction to GMR and SVR highlighting pros and cons through a set of qualitative examples (2D drawings made by hand) (1-3 slides)
- description of high-dimensional dataset you are working with (e.g. source images + projections, peculiarities) (1 slide)
- range of values for parameters chosen for the systematic comparison (1 slide)
- qualitative + quantitative results (2-4 slides)
- conclusion and take-home-message (1 slide)

Of course this is just a starting point and might not be optimal for your specific presentation. Plan your presentation as if your audience has a fair understanding of machine learning, but has NO IDEA of what you did during the practicals or what algorithm parameters you were able to play with. This is important as it will help you make a more general discussion of the methods you studied without focusing only on the practical aspects of what the software did.

#### 6.1 Format

As a general rule, the presentation should contain both a **qualitative** and a **quantitative** estimation of the performance of the system:

A **qualitative evaluation** should contain images (e.g. screenshots) which exemplify the concepts you want to explain (e.g. an image of a good classification and an image of a bad one). Make sure to plot only a subset of all the plots you may have visualized during the practical. Choose the ones that are most representative. Make sure that there is no redundancy in the information conveyed by the graphs and thus that each graph presents a different concept.
The **quantitative evaluation** should contain graphs and tables of the training/testing errors. You should ALWAYS include mean and standard deviation of your observations (preferably in the form of mean ± stdev). If you cannot compute this (e.g. you only have one single performance value) you have done something wrong: rinse and repeat. Computing the standard deviation of your results also allows you to get an insight of how accurate the system is and to make reasonable comparisons (e.g. a method with an accuracy of 95% ± 10% is not necessarily better than a method with an accuracy of 94% ± 1%).

The titles, axes and legends must be clearly explained in the graph or in the caption: unlike several reptiles and birds, human eyes do not have a zoom capability: be sure to write your axes, parameters, legends and tables in a properly sized font. The images below show an example of what you should and shouldn’t do. A penalty will be given for each improperly defined graph!

![Classification Performance of Face Images](image1.png)

![Classification Performance of Face Images](image2.png)

Figure 4: Examples of good, bad and worse ways of presenting your graphs. **LEFT:** the title and axes are readable, the legend is clear. **CENTER:** the title is there but is hardly readable, the axes and parameter names are too small to see. **RIGHT:** the axes are unreadable, we have no idea what the graph is representing, there is no legend, no standard deviation on the graphs themselves: graphs presented this way will be penalized.

### 6.2 Grading scheme

The presentation will be graded on the following considerations:

- Explanation of the algorithms used (30%).
- Clarity of presentation of the qualitative and quantitative results. (30%)
- Interpretation of the qualitative and quantitative results. (30%)
- Timing of the presentation. (10%)