Machine Learning I
Practice Session Regression

Teacher: Aude Billard

This practice session will allow you to compare the two main nonlinear regression techniques presented in class: Support Vector Regression (SVR), Gaussian Mixture Regression (GMR) For SVR, we consider only the RBF kernel, so that it is easier to compare with GMR. We briefly summarize the algorithms in the next section.

To compare the algorithms, we use 2-dimensional datasets for visualization purpose. We compare the sensitivity of the algorithms to the presence of noise, missing data and choice of hyperparameters to lead to overfitting.

1 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. $\hat{y} = f(x) = \sum_i \alpha_i k(x^i, x)$, with α_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, ϵ, 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 12).

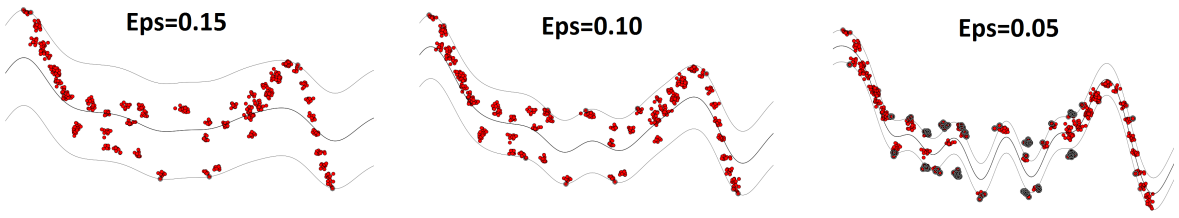

Figure 1: regression using svr. the image shows different values for the ϵ parameter, a *tube* is drawn around the mean corresponding to $f(x) \pm \epsilon$. a large tube does not need to be very complex in shape to contain most samples, whereas a smaller one will result in a more complex shape.

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 X and output data Y, 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, i.e. $\mathbb{E}[p(y|x)]$. In other words:

- $p(y, x)$ is the joint probability density function of (X,Y) and is modeled using GMM (Expectation-Maximization).
- $p(y|x)$ is the conditional probability density function of Y given the occurence of the value x of X and is estimated from the trained GMM.
- $\hat{y} = \mathbb{E}[p(y|x)]$ is the regression function (or regressive signal) and is computed in GMR by taking the expectation of the estimated conditional probability density function.
- $\text{Var}\{p(y|x)\}$ is the variance associated to each regressive output and can additionally be estimated from the GMM model.

Figure 2 shows an example of the GMR output given by MLDemos. The central line is the regressive signal, whereas $\pm$ 1 and $\pm$ 2 standard deviations are denoted by the lighter lines surrounding the regressive signal. The shading corresponds to the likelihood of the associated GMM. Recall that the variance of the regressive model $\text{Var}\{p(y|x)\}$ gives information on the relative dispersion of the predictive function. 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.

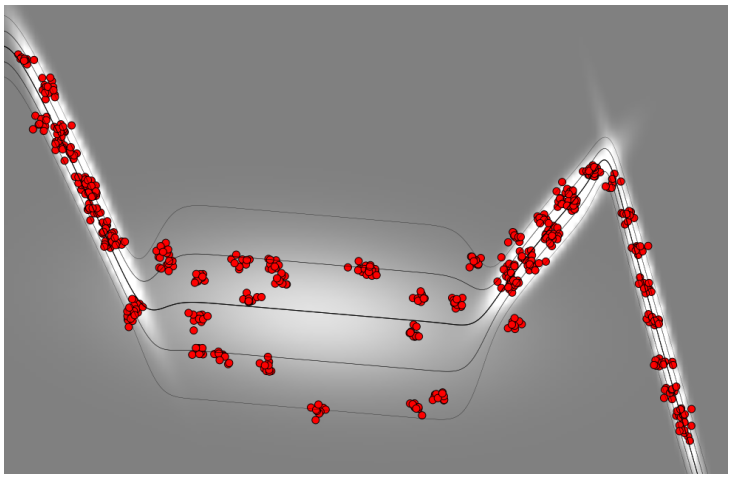

Figure 2: Regression using GMR. The grayscale background shows the likelihood in the different zones of the input/output domain. This allows to estimate how confident the output of the regression is.

3 TODO for this practice session

3.1 Datasets

In this practice session, we will ask you to work on 3 datasets which we already prepared for you. You can find them on Moodle. Download the dataset and load each example by a simple drag-and-drop of the file in the canvas of MLDemos or using the `Load` function from in the Menu bar (`File->Load`).

The three files are:

`Example_overfit.ml`: In this example, the data show very small oscillations. If the algorithm fits these oscillations perfectly, this will lead to overfitting with large error on the testing set. You will determine which parameters lead to overfitting for both SVR and GMR.

`Example_state_dependent_noise.ml`: In this example, you have a dataset where the noise becomes twice bigger as you move in the positive value for x. SVR does not adapt the ϵ parameter to the noise, whereas GMR adapts the variance to fit better the noise. You will compare qualitatively the two techniques.

`Example_missing_data.ml`: In this example, part of the data was removed, creating a region with no datapoint. This simulates an example in which you would have lost samples from your recordings. Nonlinear regression methods interpolate differently away from the data. You will grow an understanding of the interpolation done by SVR and GMR.

`Example_imbalanced_dataset.ml`: In this example, the data is not distributed homogeneously. This simulates a real case scenario where you would have gathered more datapoints in one region of the state space than in another one. was removed, creating a region with no datapoint. Nonlinear regression methods depends on a cost function that gives each datapoint the same weight. Hence, they can be severely influenced by uneven spatial distribution of datapoints. However, if these large group of datapoints represent redundant information, its effect may be compensated for. You can explore this in `Example_imbalanced_dataset2.ml`.

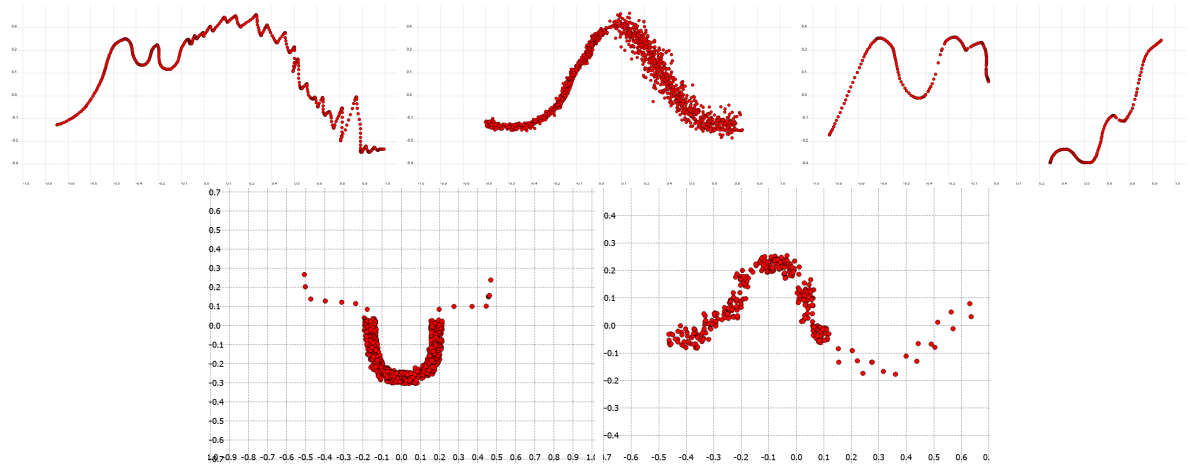

Figure 3: The datasets you will work on. From left to right, top to bottom: `Example_overfit.ml`:, `Example_state_dependent_noise.ml`, `Example_missing_data.ml`, `Example_imbalanced_dataset.ml`, `Example_imbalanced_dataset2.ml`.

3.2 Task

Overfitting

Load the dataset `Example_overfit.ml`. Vary the hyperparameters of each technique and use the compare button to assess the performance:

- For SVR: Fix $C = 1$. Vary the values of ϵ and the kernel width σ.
- For GMR: Try increasing the number of Gaussians until you see a clear overfit.

- For both techniques: Test the effect of changing the training/testing ratio by selecting values in range 10% to 90% (e.g. 10%, 25%, 66%, 90%).
  Question: For which hyperparameter, do SVR and GMR overfit?

- **Solution:**

  In general SVR tends overfit for small sigma, large C and small epsilon. Remember that overfitting behaviour is indicated by a small training error and a high testing error. We can see this for $C = 100, \epsilon = 0.0001, \sigma = 0.001$. We can show an overftting behaviour by choosing a training/test ratio of 25%. The dataset is highly dense, so a ratio of 25% is fine in this case (see Figure 4). Usually we would recommend a ratio of 50% or 66% for cross-validation. Compare this with a more robust model with parameters $C = 1, \epsilon = 0.01, \sigma = 0.01$.

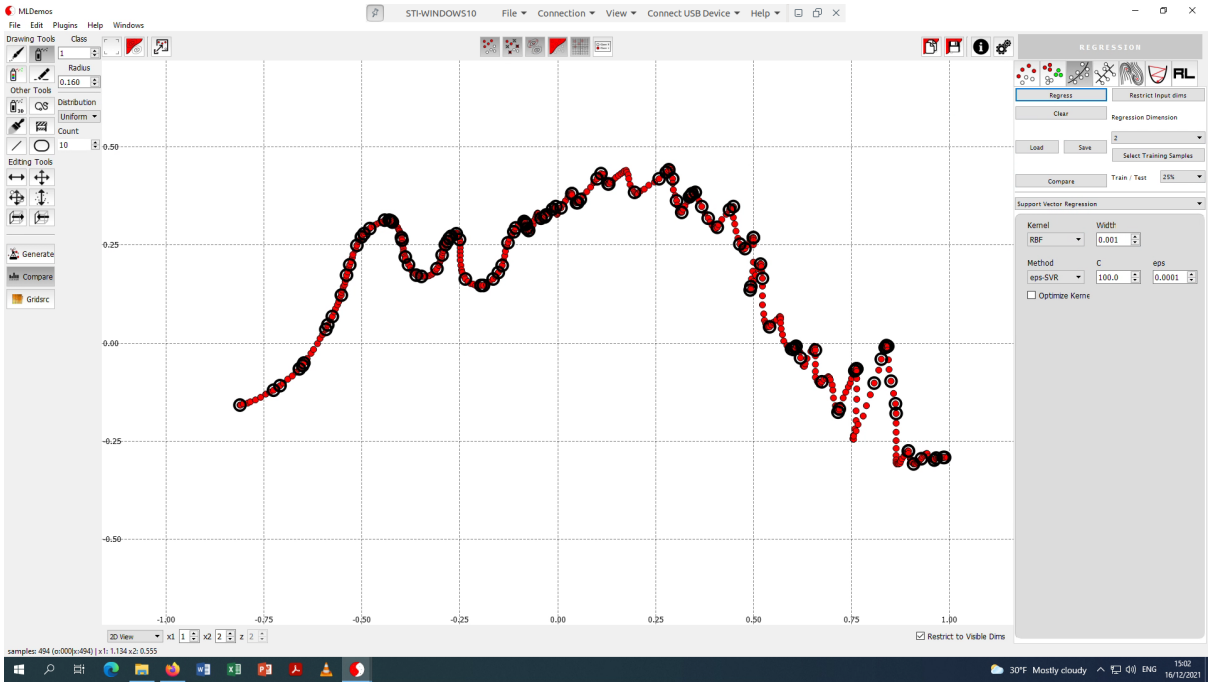

Figure 4: Visualization a 25% training set on this dataset using the support vectors for a very small ϵ.

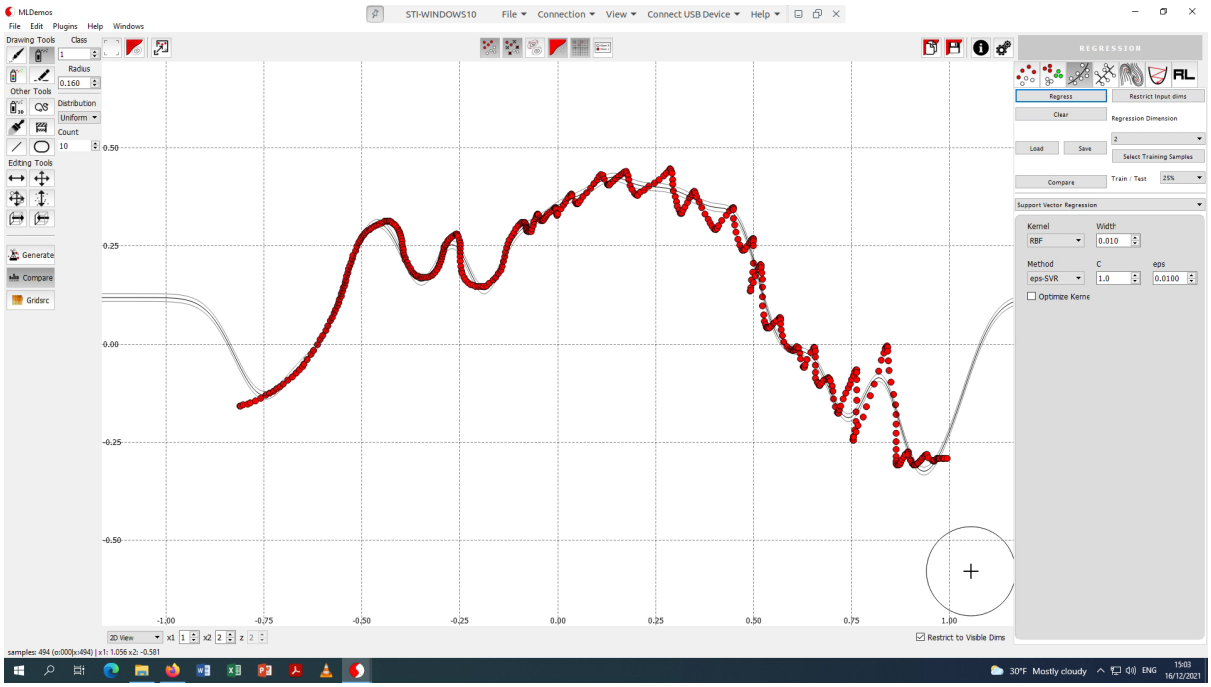

Figure 5: SVR of the stable model ($C = 1, \epsilon = 0.01, \sigma = 0.01$) 25% training set.

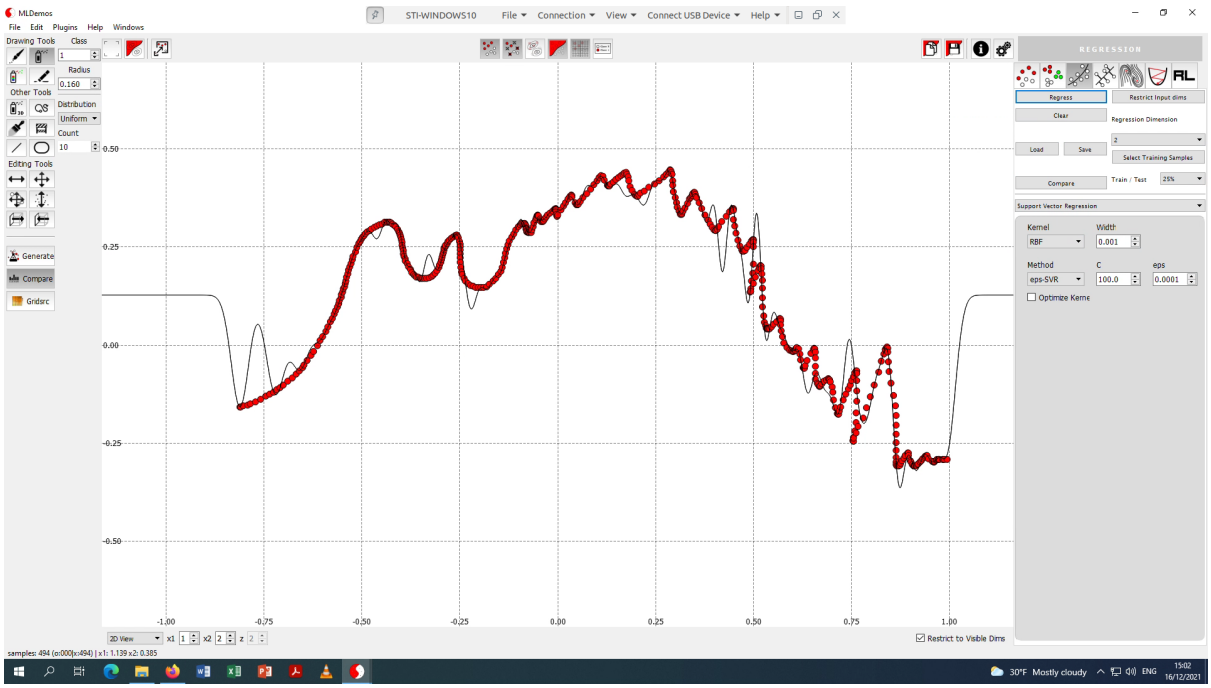

Figure 6: SVR of the unstable model ($C = 100, \epsilon = 0.0001, \sigma = 0.001$) 25% training set.

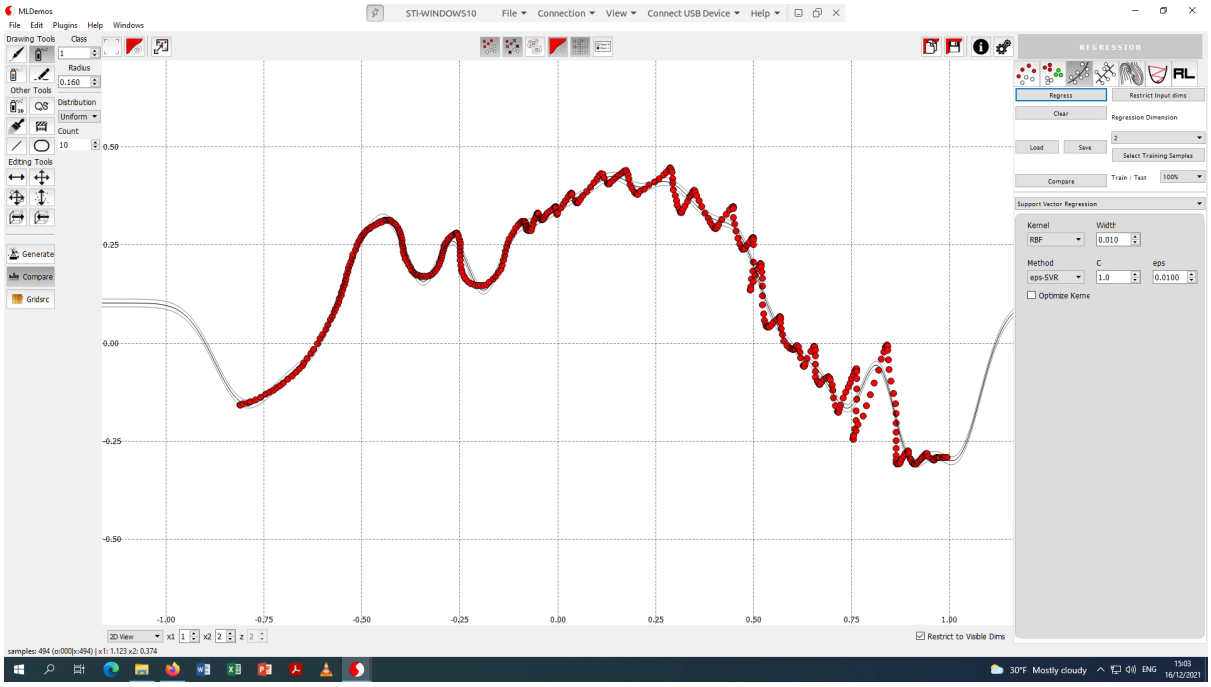

Figure 7: SVR of the stable model ($C = 1, \epsilon = 0.01, \sigma = 0.01$) 100% training set.

Noise dependent dataset

Load the dataset `Example_state_dependent_noise.ml`. Choose 100% for the training/testing ratio.

- For GMR: Select $K = 4$ with Full Covariance matrix.

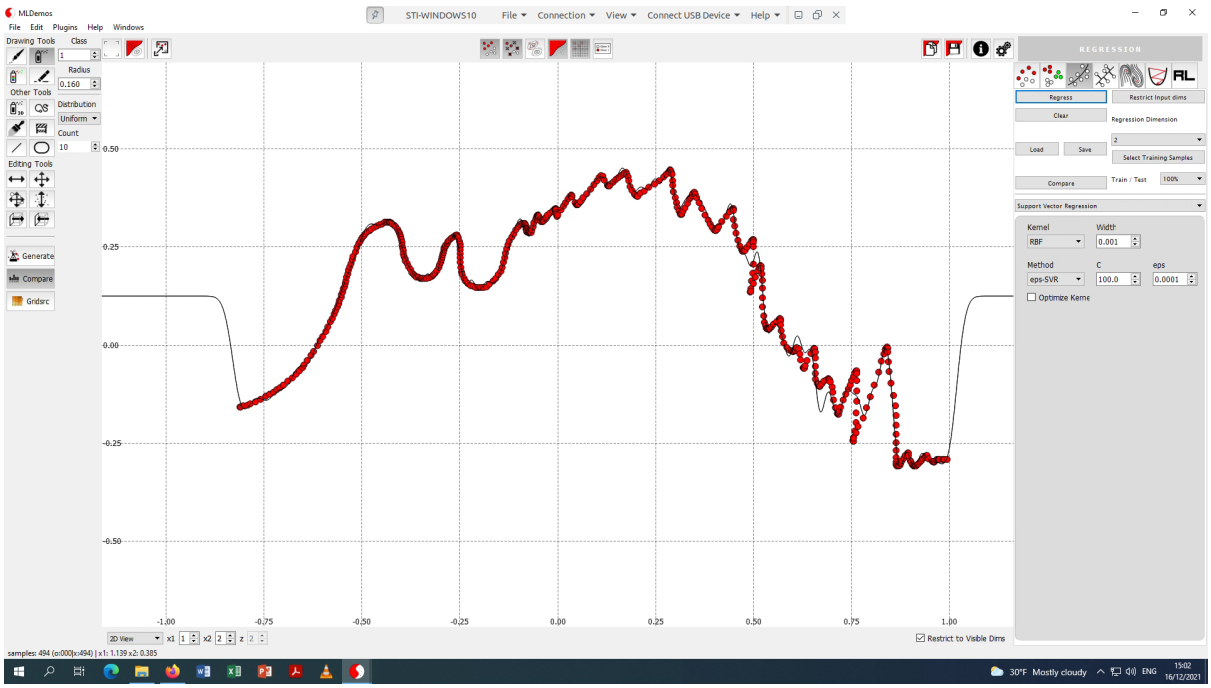

Figure 8: SVR of the unstable model ($C = 100, \epsilon = 0.0001, \sigma = 0.001$) 100% training set.

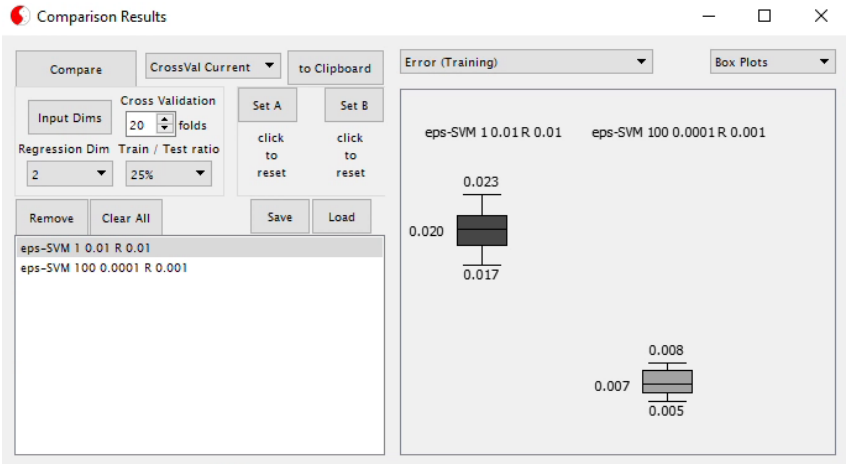

Figure 9: Cross-validation train error on the overfitting dataset.

Question: What is the effect of the noise on the fit?

The noise affects the variance of the mixture components. A larger noise results in mixture components with larger variance, thus the confidence intervals become larger. This tells us that GMR can express the noise inherent to the dataset if fitted well.

- For SVR: Select a kernel width $\sigma = 0.08$ and $C = 1$. Vary the value of ϵ.
  Question: What is the best value of ϵ?.
  For SVR a large ϵ (e.g. $\epsilon = 0.1$) underfits the left tail while a small epsilon (e.g. $\epsilon = 0.02$) fits the curve overall with smaller residual, however requires much more support vectors.

- Compare the best fit for GMR and SVR and compute the number of parameters needed

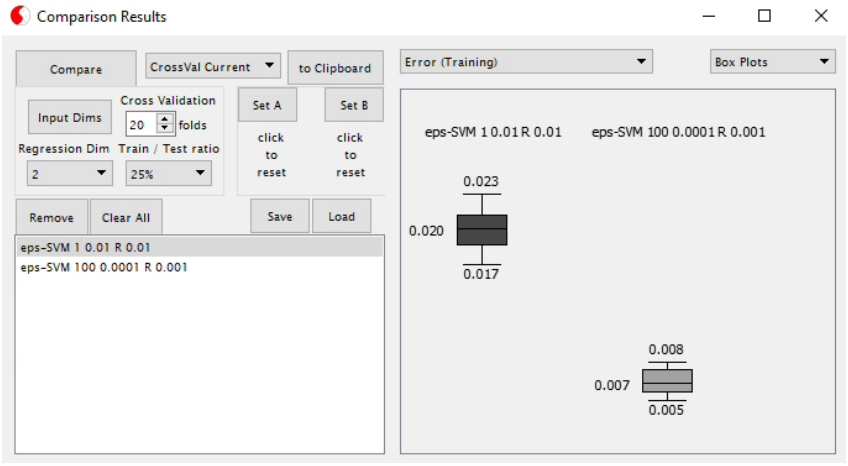

Figure 10: Cross-validation test error on the overfitting dataset.

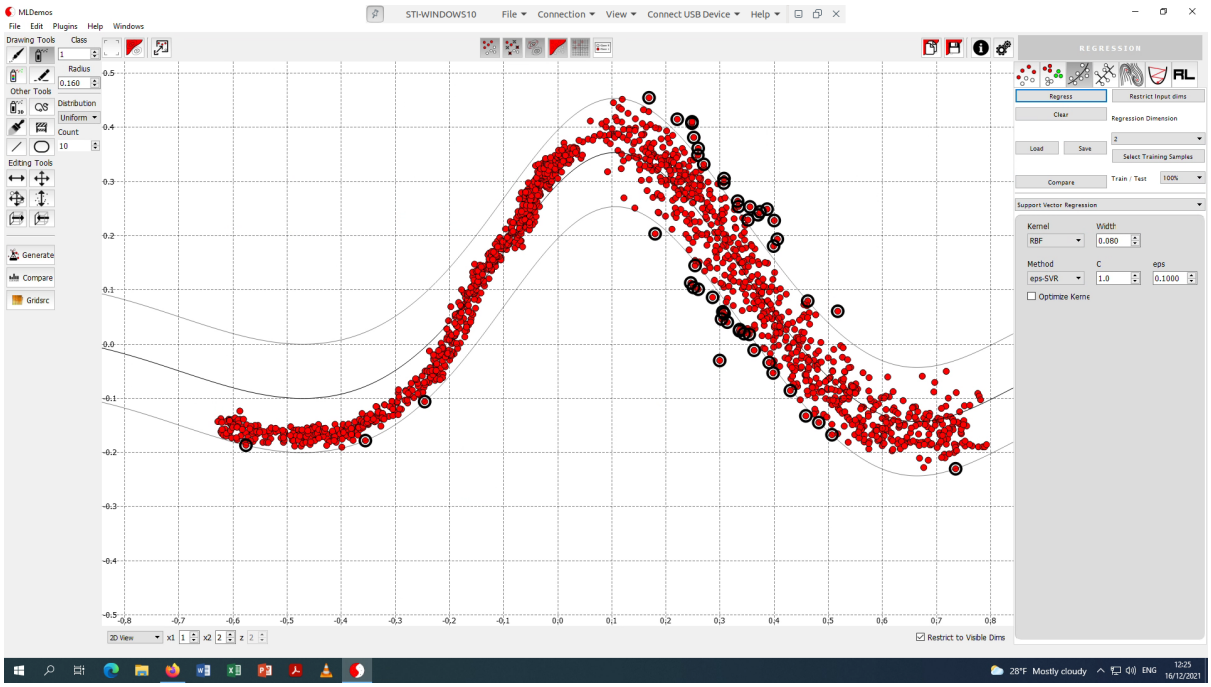

Figure 11: SVR on the noisy dataset for $\epsilon = 0.1$. One can see that the left tail is underfitted.

to store each solution.
Question: Which technique requires the fewest parameters?
GMR with $K = 8$ gets comparable testing error as SVR for $\sigma = 0.08, C = 1$ and $\epsilon = 0.02$ and thus needs much less parameters.

Missing data

Load the dataset `Example_missing_data.ml`. Choose 100% for the training/testing ratio.

- For SVR: Fix $\epsilon = 0.03$ and $C = 1$. Vary the kernel width σ to study its effect on the interpolation in the region where data is missing.
  - Find a kernel width for which the regression line goes back to b.

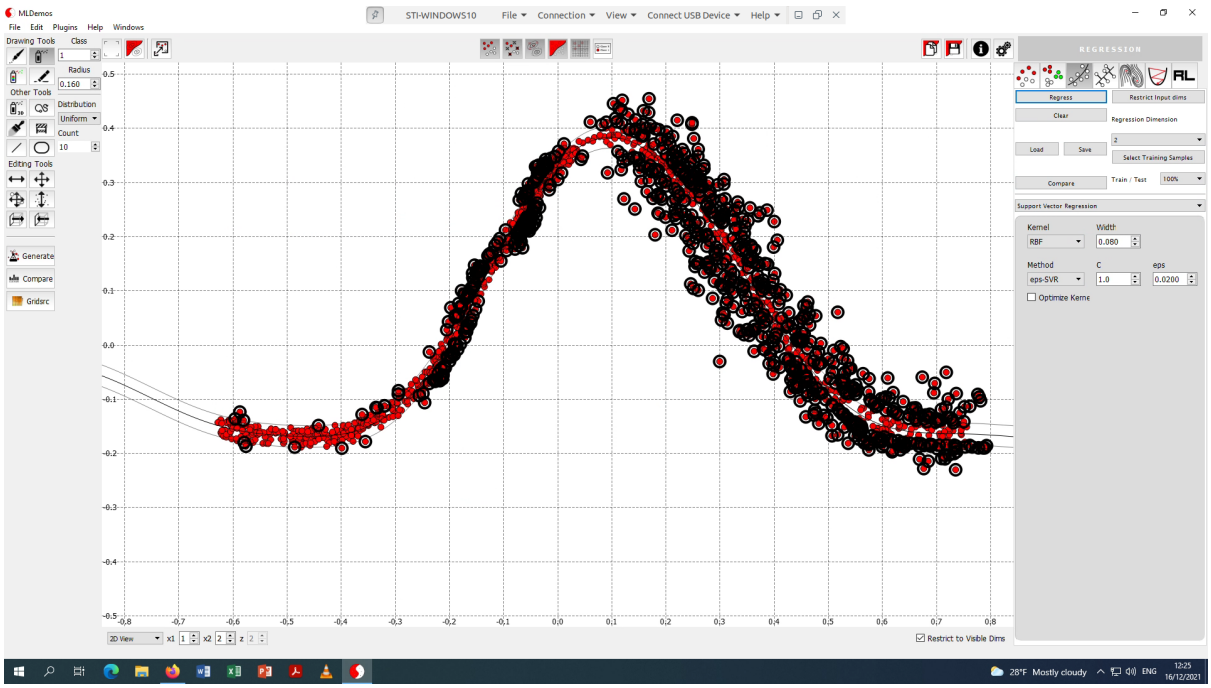

Figure 12: SVR on the noisy dataset for $\epsilon = 0.02$. One can see that it is well fitted but requires a lot of support vectors.

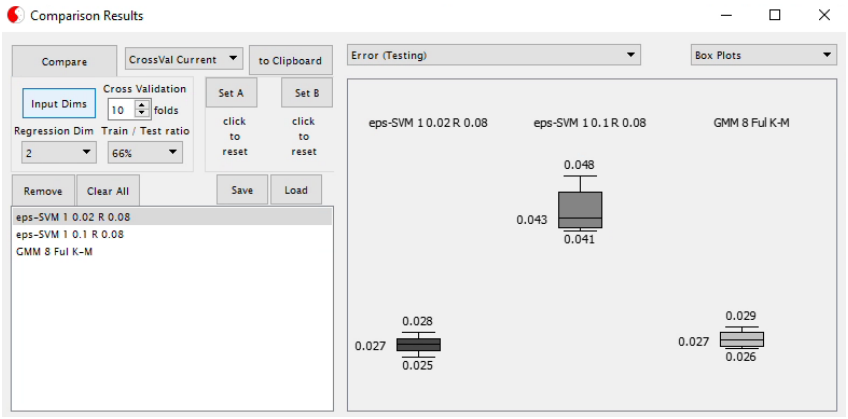

Figure 13: Cross-validation comparison for SVR and GMR on the noisy dataset.

- Find a kernel width for which the interpolation seems good to you.
- Use the compare button and assess the performance on these two choice of kernel width.

Question: For which kernel width do you get the best interpolation? Do you get good performance (in testing error) for this kernel width?
A smaller σ increases the variance of the model, and therefore increases the variance of the interpolation in the area of missing data. A larger σ smoothes the interpolation in this area. A visual good fit can be obtained with $\sigma = 0.03$, however the best testing error can be obtained with $\sigma = 0.01$, since it fits better the curve for the datapoints which are actually in the dataset.

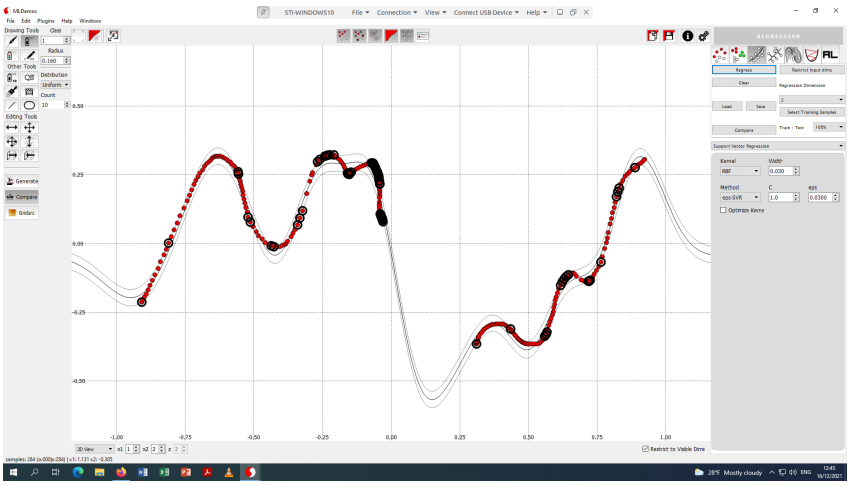

Figure 14: SVR fit on the missing dataset for $\sigma = 0.03$.

Figure 15: SVR cross-validation results on the missing dataset for $\sigma = 0.1, 0.01, 0.001, 0.03$.

- For GMR: Use Full Covariance matrix and study the effect of increasing the number of Gaussians on the interpolation. Try values in range $K = 2$ to $K = 10$. In each case, as the performance may depend on the initialization, run GMR several times and keep the best run to avoid.
  Question: For which value of K, do you get good performance? We obtain the best test error for $K = 10$. GMR starts to overfit only for very large number of clusters for this dataset (e.g. $K = 25$)

Imbalanced Datasets

Load the dataset `Example_imbalanced_dataset.ml`. Choose 100% for the training/testing ratio.

- For SVR: Fix $C = 1$. Vary the kernel width σ and ϵ to study its effect on the interpolation in the region where data is scarce.
  - Find hyperparameters for which the regression line gives the best fit for the scarce data.
  - Check how this choice affects number of SVs.

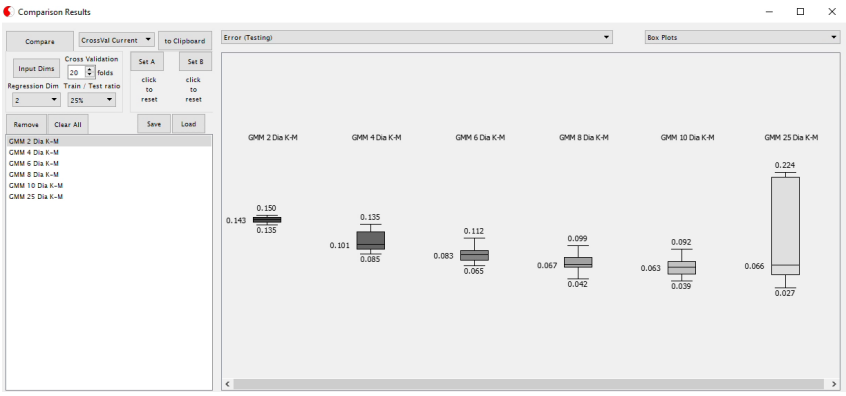

Figure 16: GMR cross-validation results on the missing dataset for $K = 2, 4, 6, 8, 10, 25$.

- For GMR: Compare type of Covariance matrix and study the effect of increasing the number of Gaussians on the interpolation for the missing data. Try values in range $K = 2$ to $K = 10$. In each case, as the performance may depend on the initialization, run GMR several times and keep the best run to avoid.

  Question: Which of GMR or SVR is most sensitive to the imbalance dataset? The performance of GMR depends highly on the initial mixture components. The mixture components tend to move to the dense areas leaving sparse regions underrepresented. One can compensate this tendancy by choosing a uniform initialization. However, for the second imbalanced dataset a uniform initialization has problems to fit fast changing points well (see the dip in Figure 17. In contrast, SVR does not have these density-related problems. Overall GMR is therefore more sensitive than SVR to the imbalanced datasets.

Redo the above with dataset `Example_imbalanced_dataset2.ml`.

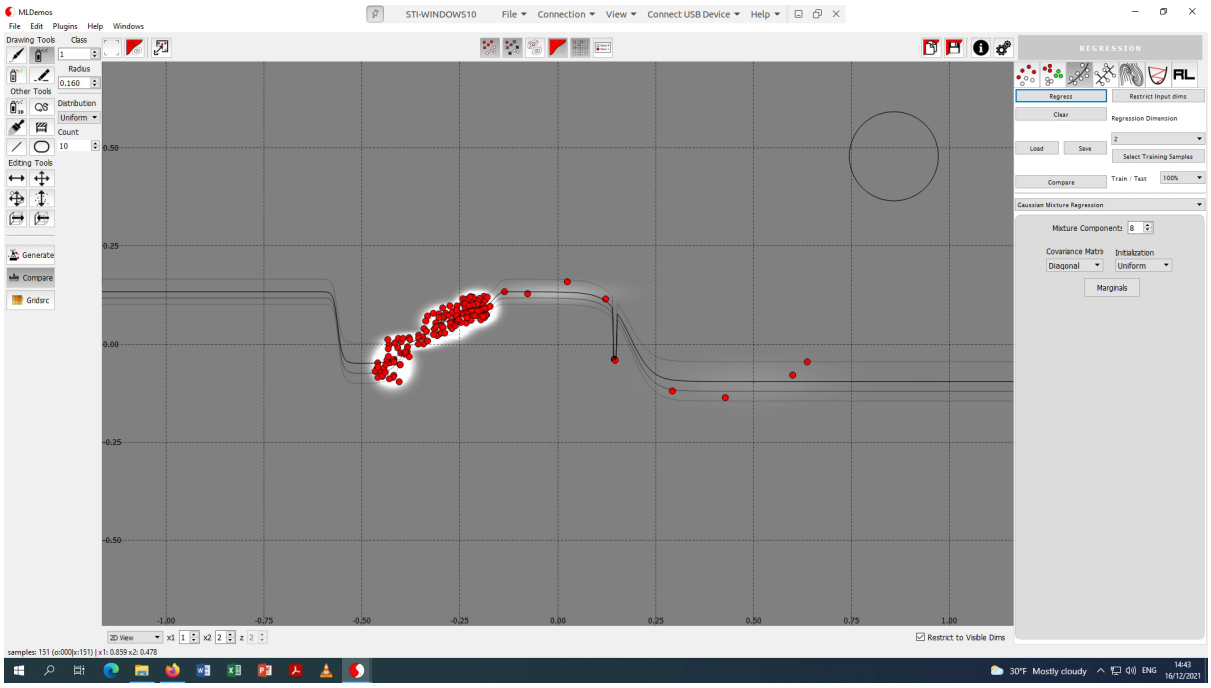

Figure 17: GMR for the second imbalanced dataset. You can see the dip at $x1 = 0.15$, $x2 = 0.05$ caused by the fast change of $x2$.