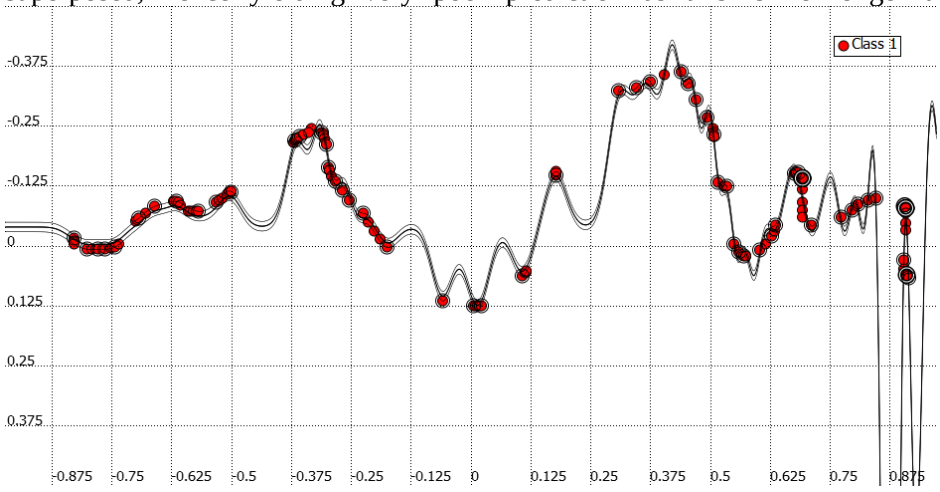
EXERCISE SESSION Nonlinear Regression: ADVANCED MACHINE LEARNING COURSE – EPFL – Lecturer A. Billard

Question 1: Support Vector Regression

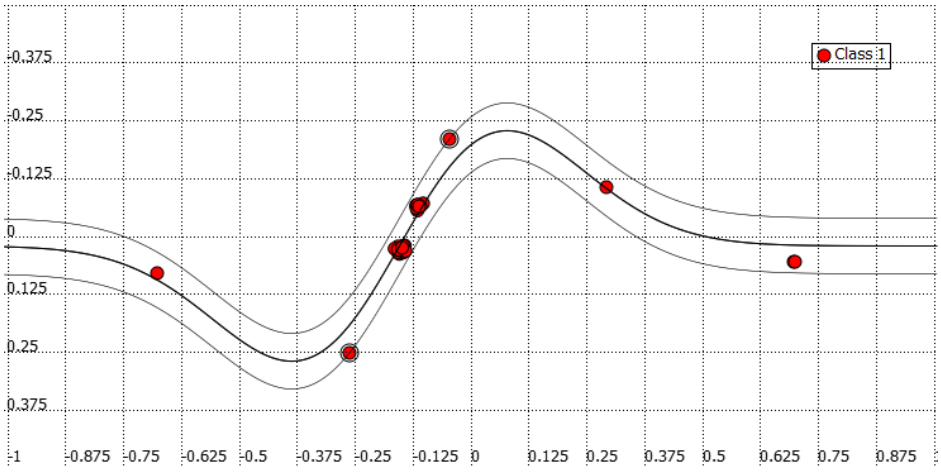
1) Show with a 2-dimensional schematic (where the first coordinate is used to predict the second coordinate, i.e. $x_2 = f(x_1) = \sum_{i=1}^{M} (\alpha_i^* - \alpha_i) k(x_1^i, x_1) + b$, that you can fit any combination of points when using SVR with the Gaussian kernel. How many data-points at minimum do you need as support vectors? Can the ε-tube have an effect on this minimum number of points?

Solution:

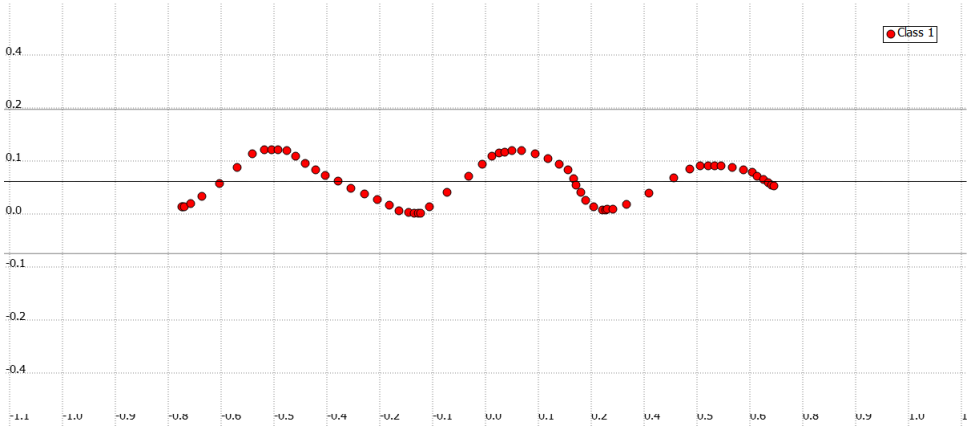
By changing the magnitude of the kernel width, one can fit arbitrarily small fluctuations of the curve. The cost is an increase in the number of SVs (and over-fitting). At worst, all datapoints become support vectors. Inference in-between datapoints will be extremely poor. The example below shows such a poor fit. The points on the far right are almost superposed, hence yielding very poor prediction as this is no longer a function.



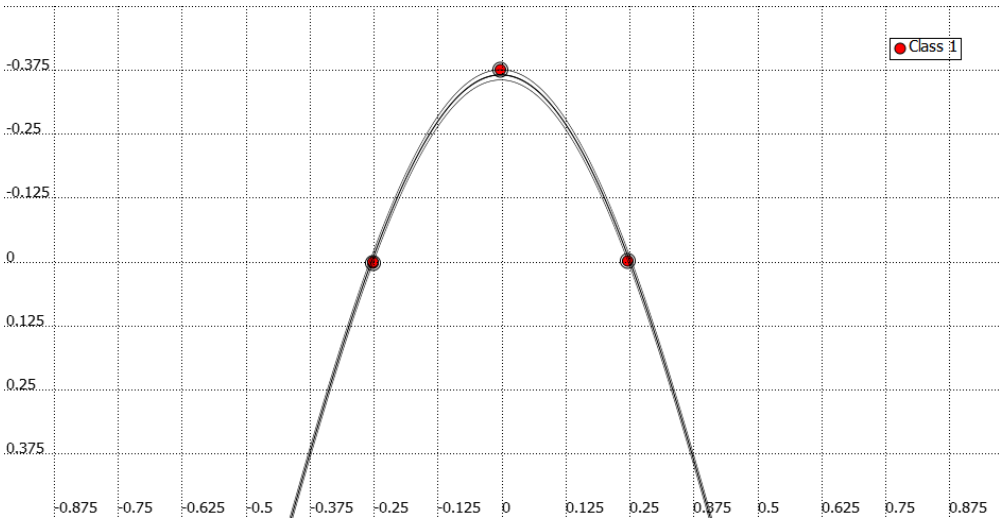
You need at minimum two SVs to satisfy the constraints that $\sum_{i=1}^{M} (\alpha_i^* - \alpha_i) = 0$ (see derivation of the Dual in the slides of the class).



If the epsilon tube is large enough and encapsulates all points, none of the points will become SVs and hence, you will end up with <u>zero</u> support vectors. If the epsilon tube encompasses all points, the solution is simply a horizontal line at coordinate $y = b$.



2) The inferred function f below was fit with Gaussian kernel and a kernel width of 0.8. What is the effect of increasing the kernel width on the function?



Solution:

Recall that $f(x)=\sum_{i=1}^{M}\left(\alpha_i-\alpha_i^*\right)k\left(x,x^i\right)+b$. We can then compute b, by averaging over the value of f on each of the datapoints, i.e.:

$$y^j=f\left(x^j\right)=\sum_{i=1}^{M}\left(\alpha_i-\alpha_i^*\right)k\left(x^j,x^i\right)+b,\ \ j=1...M$$

$$\Rightarrow b=\frac{1}{M}\sum_{j=1}^{M}\left(y^j-\sum_{i=1}^{M}\left(\alpha_i-\alpha_i^*\right)k\left(x^j,x^i\right)\right)$$

Decreasing the kernel width reduces the extent to which the SV-s influences the modulation (the 1st term on the right handside of the equation for f(x)). Hence, far from the data-points, the terms $k\left(x^i,x\right),\ i=1,2,3$, are almost zero and the algorithm predicts a value equal to the intercept. The intercept is in this case equal to the average between all the y^j, as the second term on the right handside of the equation for b vanishes as well. Hence, remember that by default the algorithm will infer simply that $f(x)=b$ away from the data.

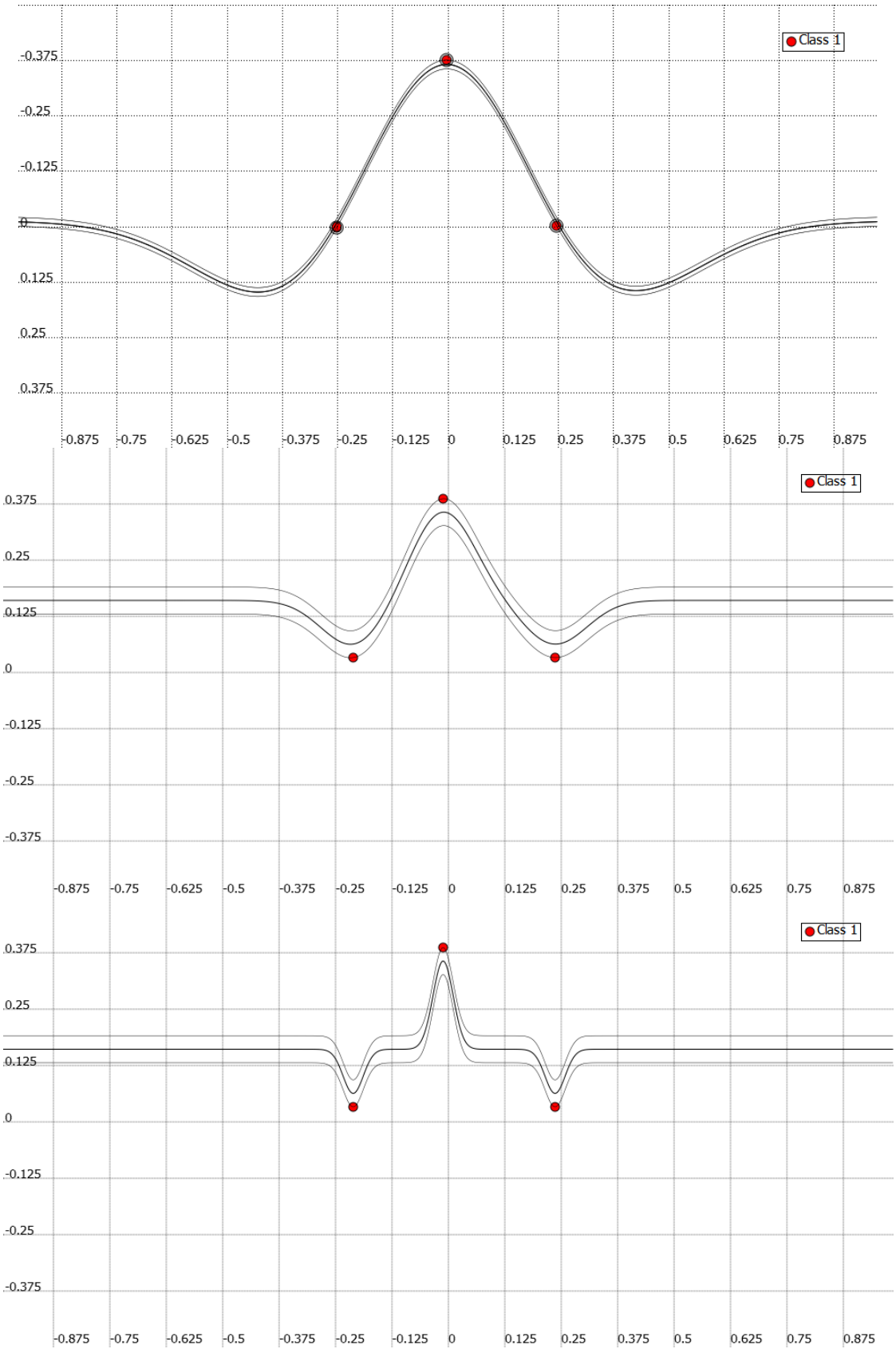
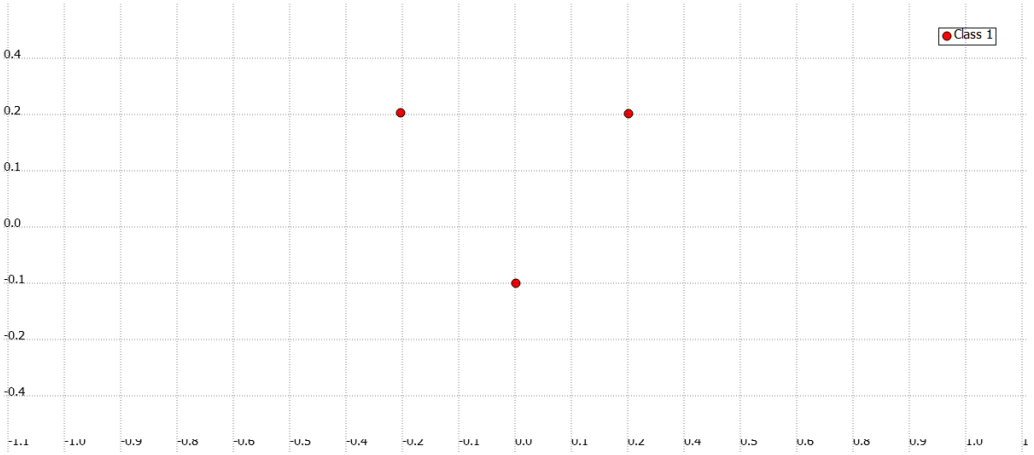


Figure 1: from top to bottom kernel width of 0.1, 0.01 and 0.001

3) What minimum order of a homogeneous polynomial kernel do you need to achieve good regression on the set of 3 points below? And how many support vectors do you need at minimum?

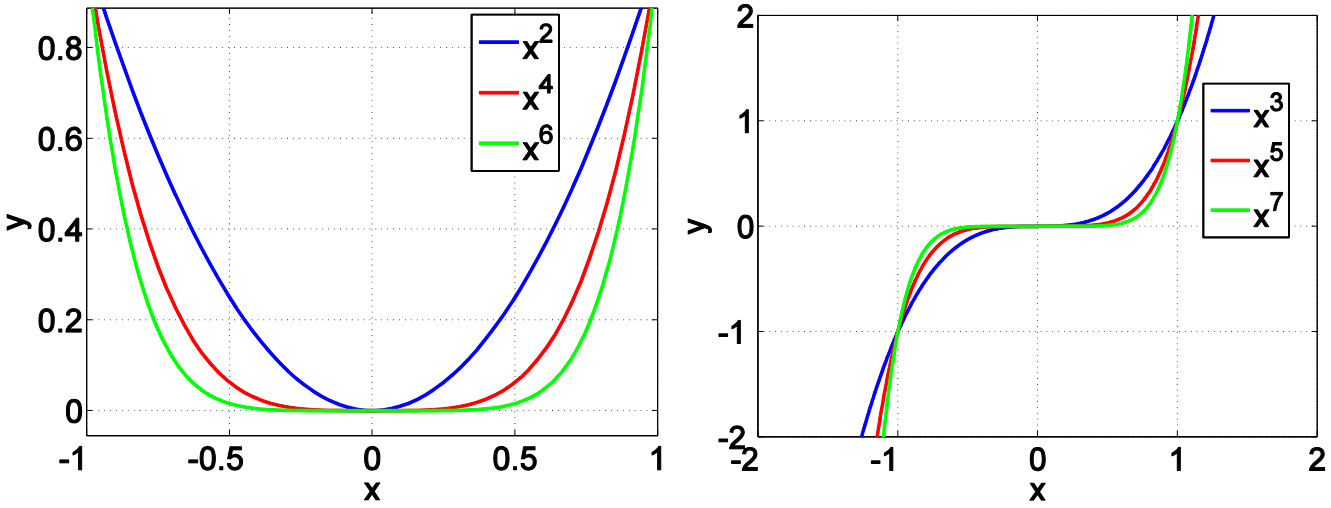


Solution:

We first identify the form of solutions that can be achieved with a degree homogeneous polynomial kernel of degree *p*. For a 1 dimensional input as we have here, the support vectors are just scalars $x_1^i \in \mathbb{R}$, where the lower index represents the coordinate of the point on the first axis. Dot products $x^T x^i$ are then reduced to just scalar multiplications resulting in the following functional form

$$y = \sum_i (\alpha_i - \alpha_i^*)(x_1^i x)^p + b : \text{ single term scaled \& shifted polynomial}$$

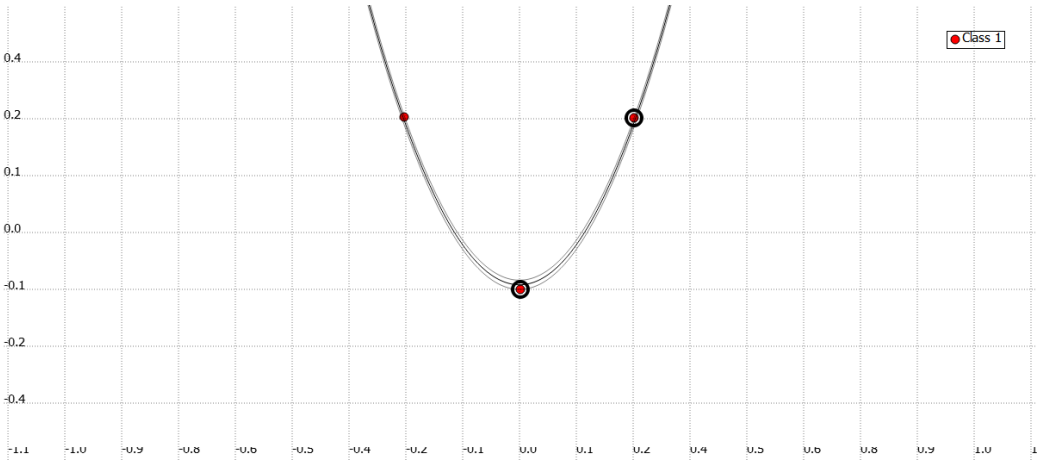
Such polynomials are easily characterized geometrically depending on whether the degree is odd or even. The following graphic shows the two possibilities.



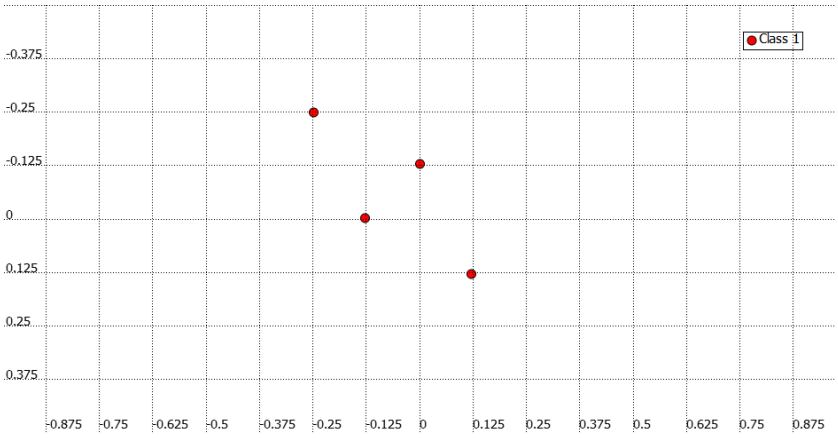
A similar effect is seen if the scaling term is increased, i.e., the function becomes steeper with the same geometrical shape. Also, having many support vectors will not change the shape since all the x_1^i are absorbed as coefficients of the testing point x.

Data sets that are distributed according to these profiles can be perfectly fitted using the corresponding kernels. For the current problem, it is immediately clear that the linear kernel will not work. Also, it is clear from the arrangement of the points that any even power kernel is suitable, minimum being the 2nd order kernel.

Since the number of support vectors does not affect the shape of the function, the minimum number of support vectors required is 2.



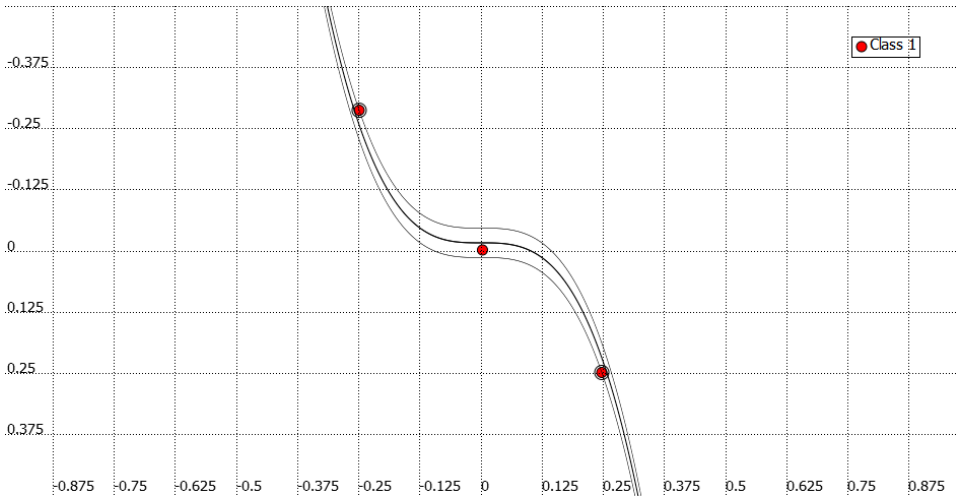
4) The dataset below cannot be fitted with polynomial kernel of order 3. Why not? Does increasing the order of the polynomial help fit these points?



5) Give examples of datasets you can fit with polynomial kernel of order 3.

Solution:

4) and 5) Polynomial kernel of order 3 (and any odd order) can fit solely these type of distributions:



As already shown above, elevating the power has no influence. Using even power does not help either since these can fit only quadratic functions.

However, inhomogeneous polynomials can produce fit of arbitrary complexity. This is due to the existence of terms raised to many powers (not just *p*, as in the homogeneous case).

$$y = \sum_i (\alpha_i - \alpha_i^*)(xx_1^i + 1)^p + b = a_p x^p + a_{p-1} x^{p-1} + \ldots + a_0 + b$$

For some scalar constants a_i.

Exercise 2: Gaussian Process Regression and Gaussian Mixture Regression

Show analytically that GPR and GMR can become equivalent under certain restrictions.

Solution:

The expression found through GPR

$$y^* \sim f^* = \mathrm{E}\left[f^* \mid x^*, X, y \right] = \sum_{i=1}^{M} \alpha_i k(x^*, x^i)$$

$$\text{with } \alpha = \left[K(X,X) + \sigma^2 I \right]^{-1} y$$

is somewhat very similar to the one found for Gaussian Mixture Regression

$$y^* = E\left\{ p\left(y \mid x^* \right) \right\} = \sum_{k=1}^{K} w_k\left(x^*\right) \cdot \mu^k\left(x^*\right) = \sum_{k=1}^{K} w_k\left(x^*\right) \cdot \left(\mu_Y^k + \Sigma_{YX}^k \left(\Sigma_{XX}^k \right)^{-1} \left(x^* - \mu_X^k \right) \right)$$

The non-linear term $k\left(x^*, x^i\right)$ is equivalent to the non-linear weights $w_k\left(x^*\right)$, whereas the parameters α_i somewhat correspond to the linear terms stemming from local PCA projections through the cross-covariance matrices of each Gaussian in GMM given by $\mu_Y^k + \Sigma_{YX}^k \left(\Sigma_{XX}^k \right)^{-1} \left(x - \mu_X^k \right)$.

The difference between GPR and GMR lies primarily in the fact that GP uses all the datapoints to do inference, whereas GMM performs some local clustering and uses a much smaller number of points (the centers of the Gaussians) to do inference. However, the two methods may become equivalent under certain conditions.

Assume a normalized Gaussian kernel for the function $k\left(x^*, x^i\right)$ and a noise free model, i.e. $\sigma = 0$. Let us further assume that for a well-chosen kernel width, $k\left(x^j, x^i\right)$ is non-zero only for data points deemed close to one another according to the metric $k\left(.,.\right)$ and is (almost) zero for all other pairs. As a result, the matrix K is sparse. We can hence define a partitioning of the datapoints through a set of $m << M$ (not necessarily disjoint) clusters $C^l : \left\{ x^i \in X,\ s.t.\ k\left(x^i, x^l\right) > \delta \right\},\ l = 1...m$, centered on m datapoints $x^l \in X$. $\delta > 0$ is an arbitrary threshold that determines the closeness of the datapoints.

Rearranging the ordering of the datapoints so that points belonging to each cluster are located on adjacent columns and duplicating each column of datapoints for points that belong to more than one cluster one can create the following block diagonal Gram matrix:

$$K = \begin{bmatrix} \left[K^1\right] & 0 & ..... & 0 \\ ..... & & & \\ 0 & ..... & & \left[K^m\right] \end{bmatrix}$$

where the elements $K_{ij}^l = \left[k\left(x^{l,i}, x^{l,j}\right) \right]_{ij}$ of the K^l matrix are composed of the kernel function applied on each pair of datapoints $\left(x^{l,i}, x^{l,j}\right)$ belonging to the associated cluster.

Using properties for the inverse of a block diagonal matrix, we obtain a simplified expression for:

$$\alpha \sim \frac{1}{\prod_{i=1}^{m} \left|K^i\right|} \begin{bmatrix} \left[\left(K^1\right)^{-1}\right] & 0 & ..... & 0 \\ ..... & & & \\ 0 & ..... & & \left[\left(K^m\right)^{-1}\right] \end{bmatrix} \begin{bmatrix} y^1 \\ . \\ . \\ y^m \end{bmatrix}$$

$$\alpha^l = \frac{1}{\prod_{i=1}^{m} \left|K^i\right|} \left[K^l\right]^{-1} y^l, \quad l = 1...m$$

Where y^l is composed of the output values associated to the datapoints X^l.

For each cluster C^l, one can then compute the centroid $\mu^l = \left\{ \mu_x^l, \mu_y^l \right\}$ of the cluster and a measure of the dispersion of the datapoints associated to this cluster around the centroid, given by

the covariance matrix $\Sigma_{xx}^{j} = E\left\{\left(X^{l} - \mu_{x}^{l} I\right)\left(X^{l} - \mu_{x}^{l} I\right)^{T}\right\}$ of the matrix X^{l} of datapoints associated to the cluster. Furthermore, for each set of datapoints X^{l}, one can use the associated set of output value y^{l} and compute the cross-covariance matrix $\Sigma_{yx}^{l} = \left(\left(y^{l}\right)^{T} - \mu_{y}^{l} I\right)\left(X^{l} - \mu_{x}^{l} I\right)^{T}$.

If we further assume that each of local kernel matrix is approximatively a measure of the covariance, i.e. $\left[K^{l}\right]^{-1} \sim \left[\left(X^{l} - \mu_{x}^{l} I\right)^{T}\left(X^{l} - \mu_{x}^{l} I\right)\right]^{-1}$ and $k\left(x^{*}, X^{l}\right)^{T} \sim \left(X^{l} - \mu_{x}^{l} I\right)^{T} x^{*}$.
Replacing in the original equation (GPR) shown above yields:

$$f^{*} = \frac{1}{\sum_{i=1}^{m}\left|K^{i}\right|}\sum_{l=1}^{m}\left[K^{l}\right]^{-1} y^{l} k(x^{*}, X^{l})$$

Observe that our prediction is now a non-linear combination of *m* linear projections of the datapoints through $\left[K^{l}\right]^{-1} y^{l}$. If the number of cluster *m* is composed of a single datapoint, we obtain a degenerate Gaussian Mixture Model with a unitary covariance matrix for each Gaussian.

Similarly, when the clusters are disjoint, the prediction f^{*} can be expressed as the product of the conditional on each cluster separately and the equivalence with GMM is immediate. In the more general case, when the clusters contain an arbitrary number of datapoints and are not disjoint, the full Gaussian Process takes into account the influence from all datapoints. In a GMM the interactions across all datapoints are conveyed in part through the weighting of the effect of each Gaussian. Note also that the centers of the Gaussians are chosen so as to best balance the effect of the different datapoints through E-M.

Exercise 3: Equivalence between GPR and SVR

a) Using an rbf kernel, for what value of the parameter b in SVR, the regression value far away from the data would be the same as the mean GPR value.

b) How would the formulation in SVR change if it is required to learn the SVR function with a fixed value of b = 0 (Hint: What is the effect of b in the dual).

c) Show that if noise is not considered in both GPR and SVR, the modified SVR above (fixed b) and the GP mean regression values are equivalent.

Solution:

a) The GPR mean regression function is of the form $\sum_{i}\alpha_{i} k(x, x^{i})$, i.e., there is no bias term as it is in the SVR. For an RBF kernel, the kernel function vanishes far away from the data and the function value is only b. Hence, SVR with $b = 0$ would give the same value as GPR far from the data (which is equal to zero as well).

b) In general, b is a variable which is optimized for in the primal, resulting in a linear constraint on α_i, i.e. $\frac{\partial L}{\partial b} = \sum_i (\alpha_i - \alpha_i^*) = 0$. If b is a fixed value, then we do not set $\frac{\partial L}{\partial b} = 0$ and hence do not get the corresponding linear constraint in the dual. The dual optimization then becomes:

$$\text{minimize } \frac{1}{2}\sum_i (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)k(x_i, x_j) + \varepsilon \sum_i (\alpha_i + \alpha_i^*) - \sum_i y_i(\alpha_i - \alpha_i^*)$$

$$\text{subject to } \alpha_i, \alpha_i^* > 0$$

In effect, the resulting values of the Lagrange multipliers α_i have larger magnitudes as compared to the general SVR case, where b is set to the average function value and subsequent errors due to non-linearity are accounted for by the term $\sum_i \alpha_i k(x, x^i)$. The attached figure illustrates this fact.

c) In the absence of noise, we would want the SVR function to pass through all the data points exactly, i.e., $\varepsilon = 0$. In this case, the optimization problem above simplifies further to:

$$\text{minimize } \frac{1}{2}\sum_i (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)k(x_i, x_j) - \sum_i y_i(\alpha_i - \alpha_i^*)$$

$$\text{subject to } \alpha_i, \alpha_i^* > 0$$

Note that the only terms appearing in the objective are $(\alpha_i - \alpha_i^*)$. Also, since there are only positivity constraints on the Lagrange multipliers, the quantity $(\alpha_i - \alpha_i^*)$ is unconstrained. We rewrite $\alpha_i \equiv \alpha_i - \alpha_i^*$ and the resulting optimization is thus unconstrained:

$$\text{minimize } \frac{1}{2}\alpha^T K \alpha - y^T \alpha$$

where $\alpha = [\alpha_1, \alpha_2, ..., \alpha_M]^T$, $K = \left[k(x^i, x^j)\right]_{ij}$ and $y = [y_1, y_2, ..., y_M]^T$ is the vectorized notation.

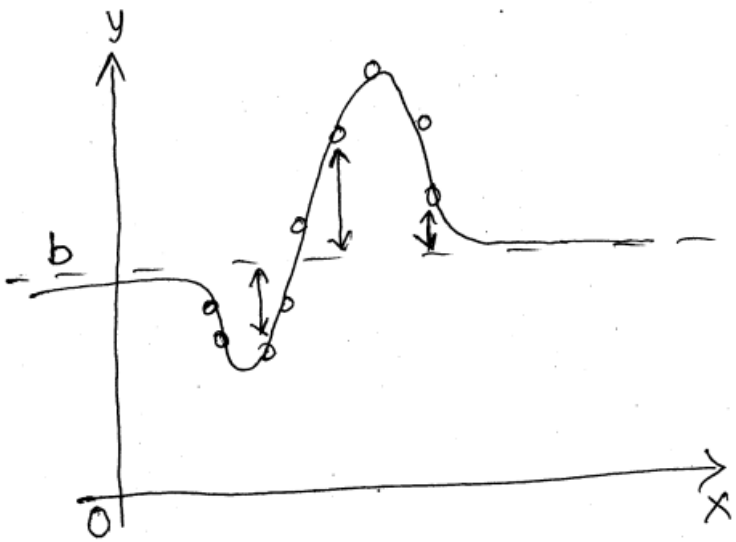
Minimizing this objective by taking derivative with respect to α and setting to zero we get

$$K\alpha = y \Rightarrow \alpha = K^{-1}y$$

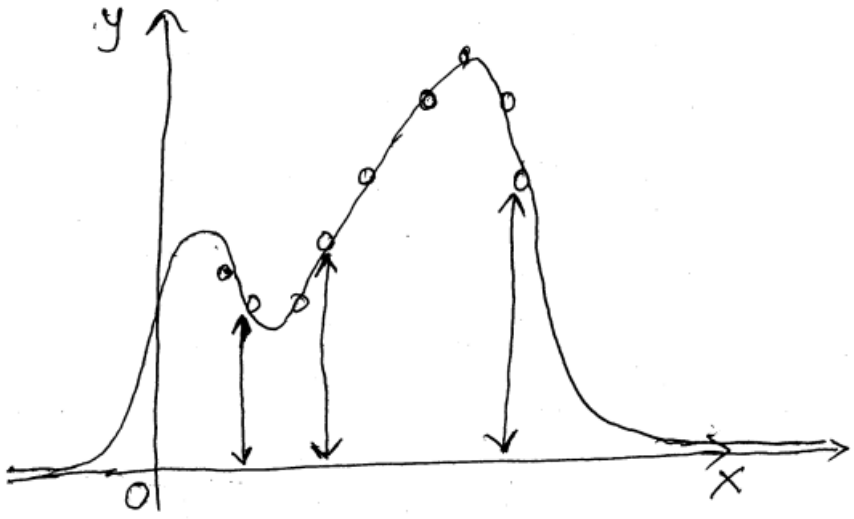
In GPR, with no noise, i.e., $\sigma = 0$, we get

$$\alpha = (K + \sigma^2 I)^{-1} y = K^{-1} y$$

Hence, with no noise in the data and b forced to zero with the modification in part b), GPR and SVR are equivalent.



$b \neq 0$



$b = 0$

o Data Points.

$\updownarrow$ $\sum \alpha_i k(x, x_i)$