Kernel Ridge Regression

Methods

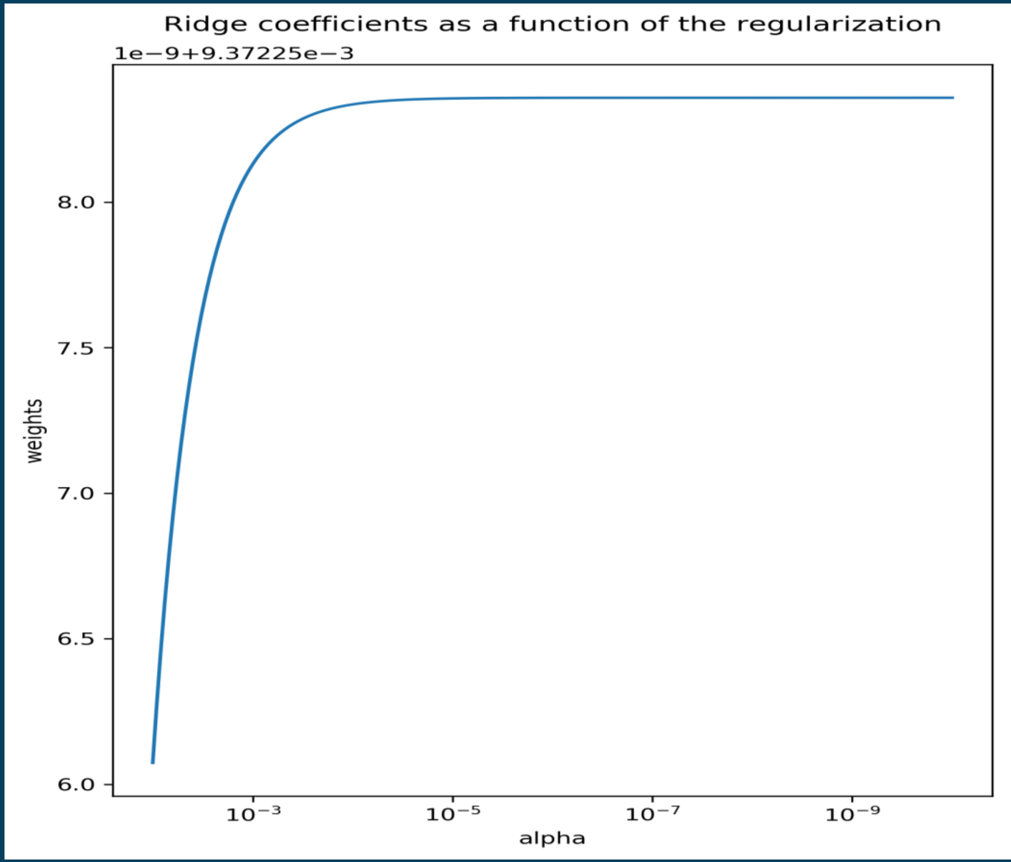
- A set of different kernels were compared and contrasted which consist of: 1- The radial basis function (RBF), 2- the Laplacian, 3- the polynomial, 4- the sigmoid and 5- the exponential sine squared (ESS) kernels

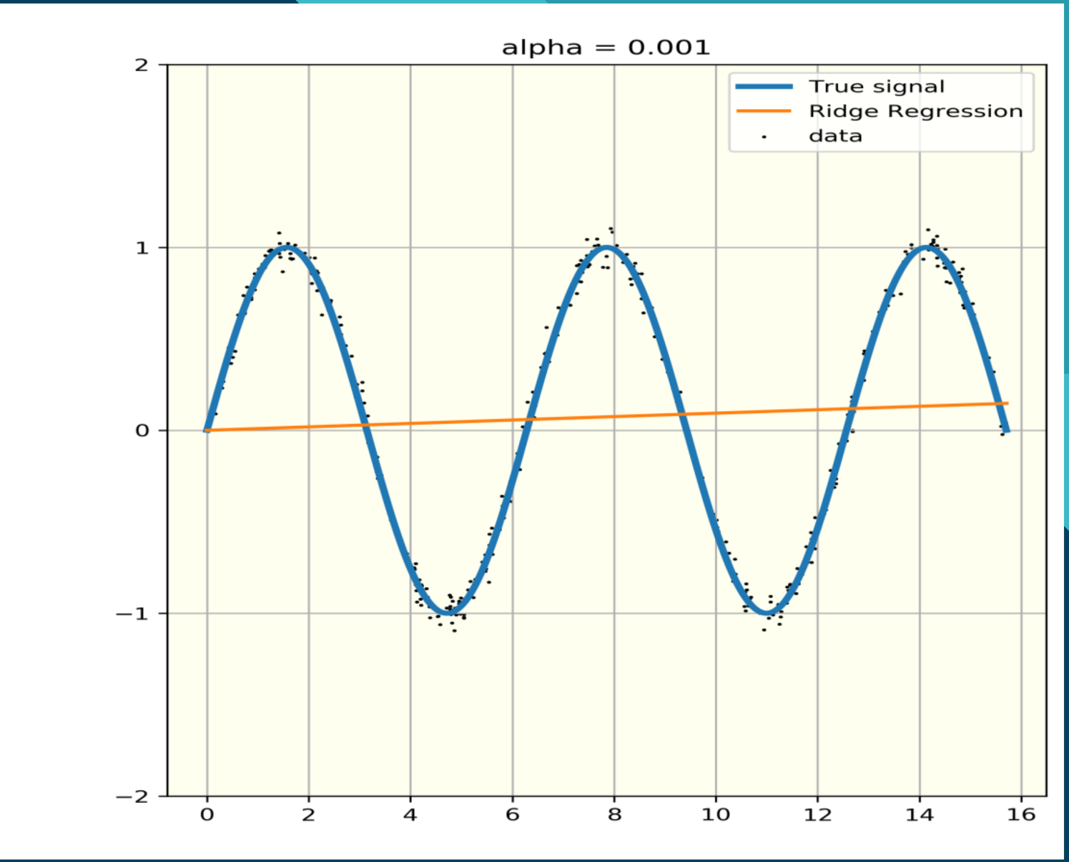
- A set of different 2D nonlinear mappings involving trigonometric functions were used to evaluate each kernel; 500 data samples (uniformly distributed random variables within a range) were used for each function and a suitable noise standard deviation was chosen to yield a SNR > 1.

- For each kernel a parameter search was done over a range of given values using sklearn's GridSearchCV function.

- 5-fold cross-validation was used for each of the kernels chosen with different parameters to compare their performance. The default metric in sklearn is the R^2 score for kernel methods. I changed this to the explained variance score and if this metric was not very informative (the scores being too close to each other) the mean squared-error was used to evaluate the kernels.

Inadequacy of Ridge Regression

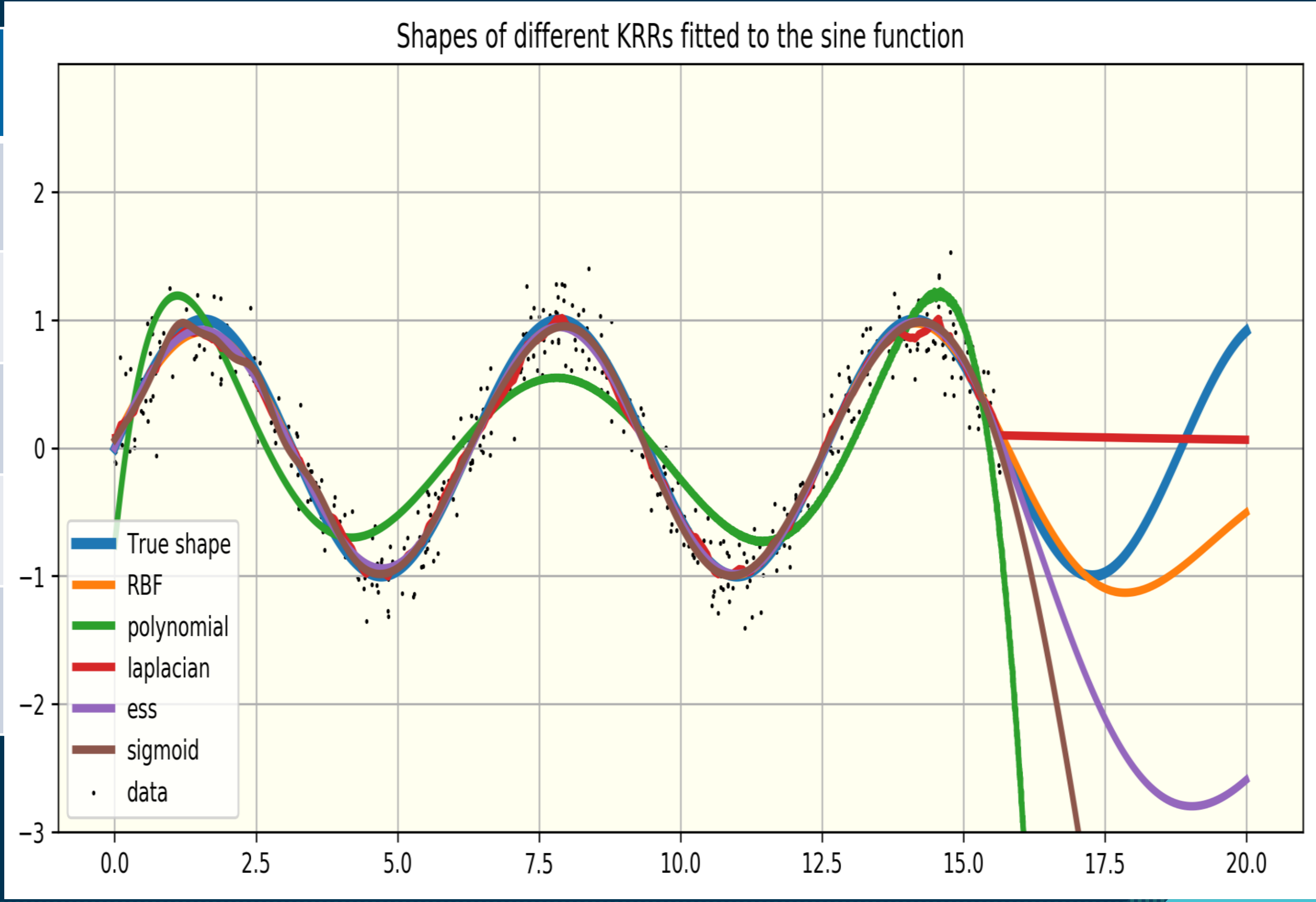




The plot to the right shows how the weight is affected by decreasing alpha and the plot to the right shows that normal ridge regression cannot be fit to nonlinear data. Since it is an extension of linear regression it has no means to handle nonlinearity.

Noise std : 0.2

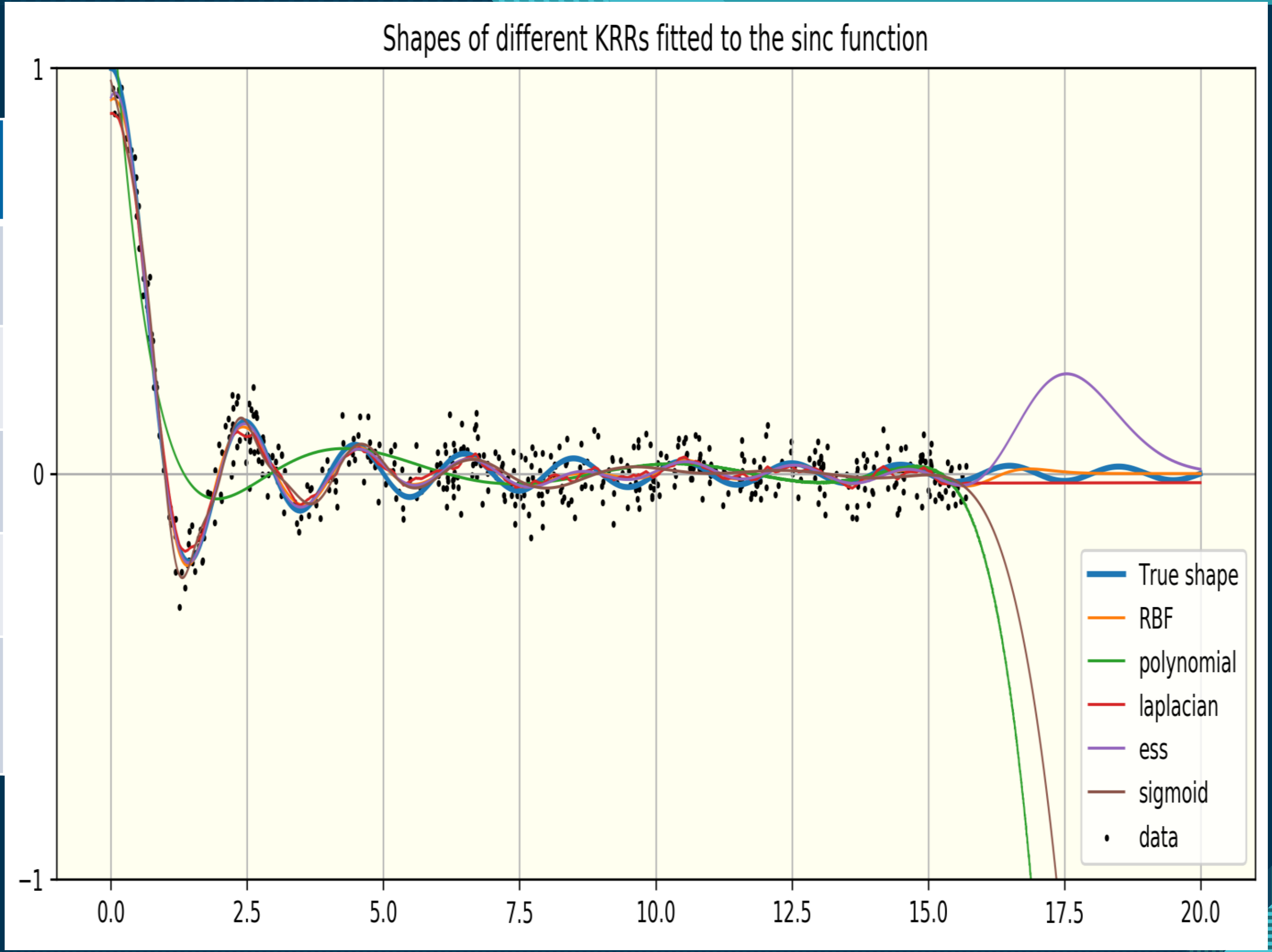
	α	γ	degree	coeff 0	CV score
RBF	0.1	0.1			0.915
Polynomial	0.1	1	6	2	0.741
Laplacian	0.1	0.1			0.908
ESS	0.01				0.915
Sigmoid	0.001	0.1		-2	0.913



For this simple sine function most kernels perform well except for the polynomial kernel which in general is not suitable for repeating patterns. In most of the figures that follow the fitted lines are continued after the domain of the data to see how they generalize.

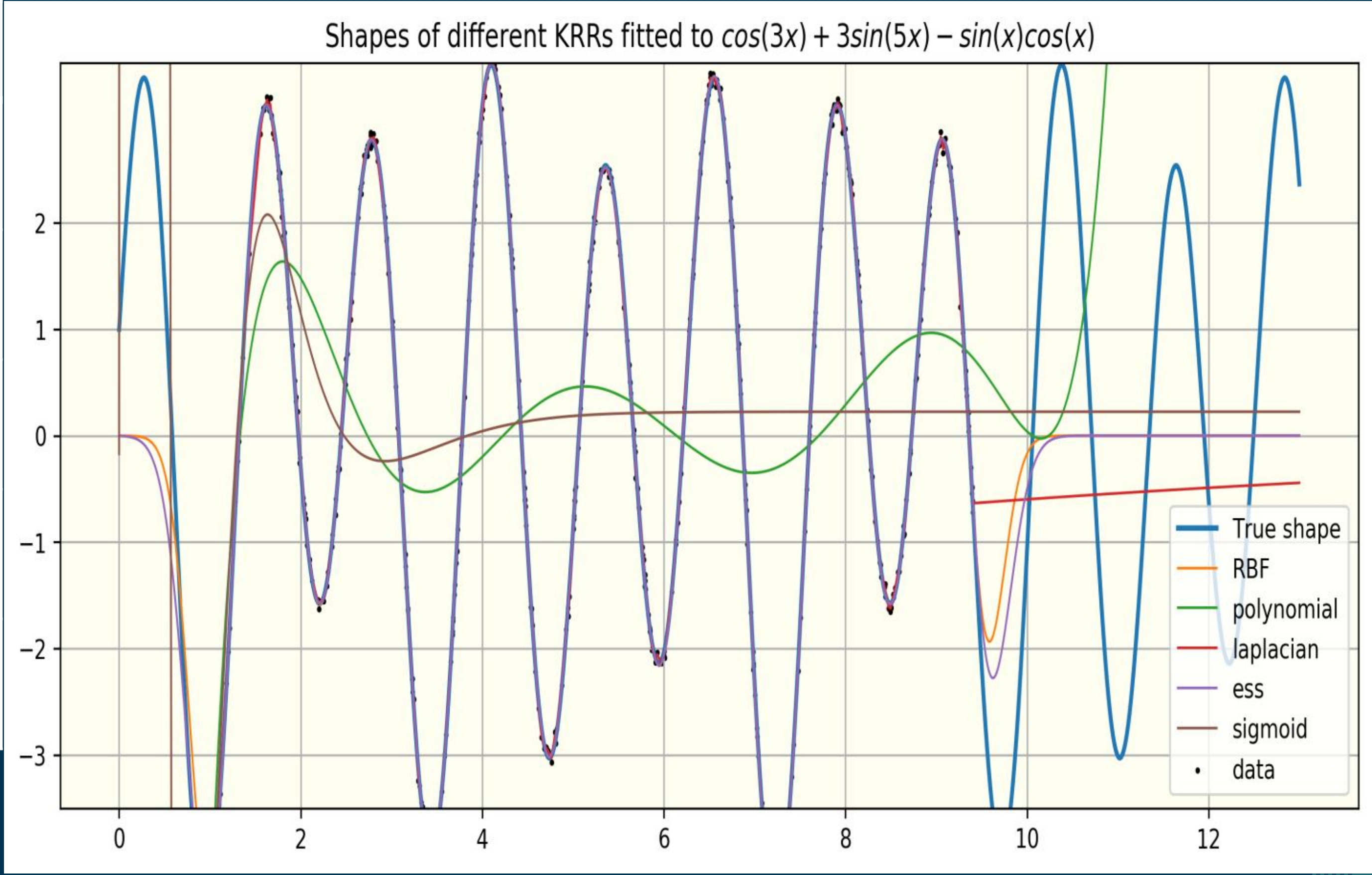
Noise std = 0.05

	α	γ	degree	coeff 0	CV score
RBF	0.1	0.1			0.851
Polynomial	0.001	0.1	7	2	0.617
Laplacian	0.01	0.01			0.835
ESS	0.001				0.854
Sigmoid	0.001	0.1		-2	0.843



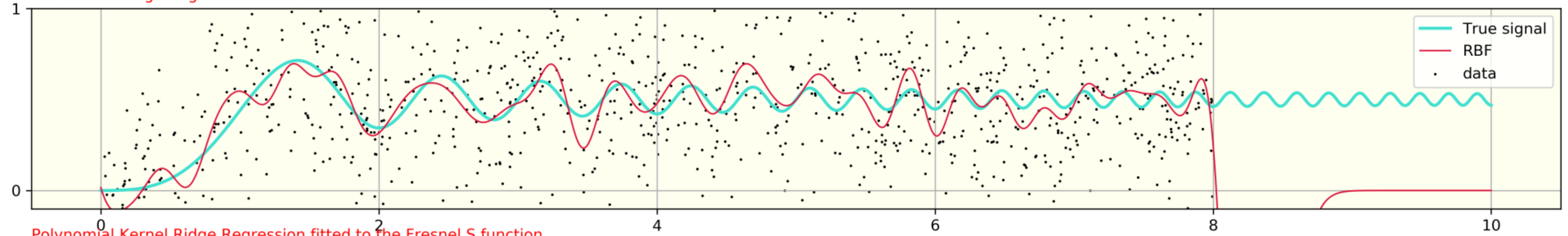
As the functions become more complicated the kernels start to perform worse, even though the noise std is lower.

	α	γ	degree	coeff 0	CV score
RBF	0.01	10			0.999
Polynomial	0.1	0.1	14	-2	0.321
Laplacian	0.001	0.1			0.998
ESS	0.01				0.999
Sigmoid	0.01	1		-1	0.07

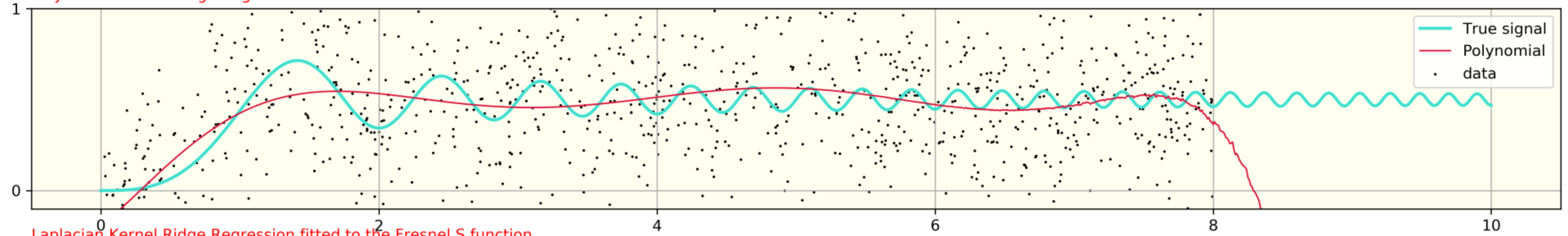


Of note here is the behavior of the sigmoid function that shows very poor performance compared to the preceding cases, perhaps due to increased periodicity of the prior data.

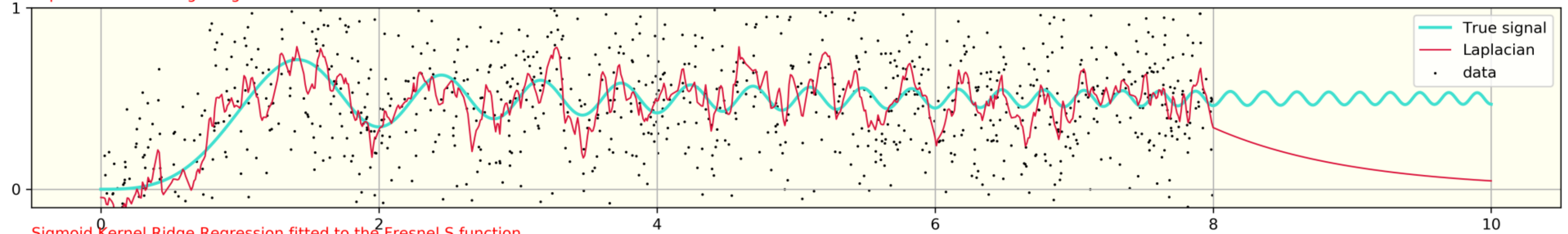
RBF Kernel Ridge Regression fitted to the Fresnel S function



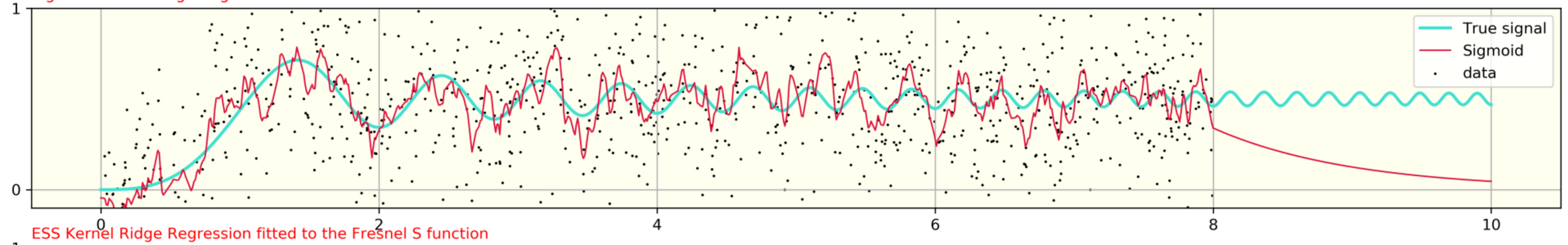
Polynomial Kernel Ridge Regression fitted to the Fresnel S function



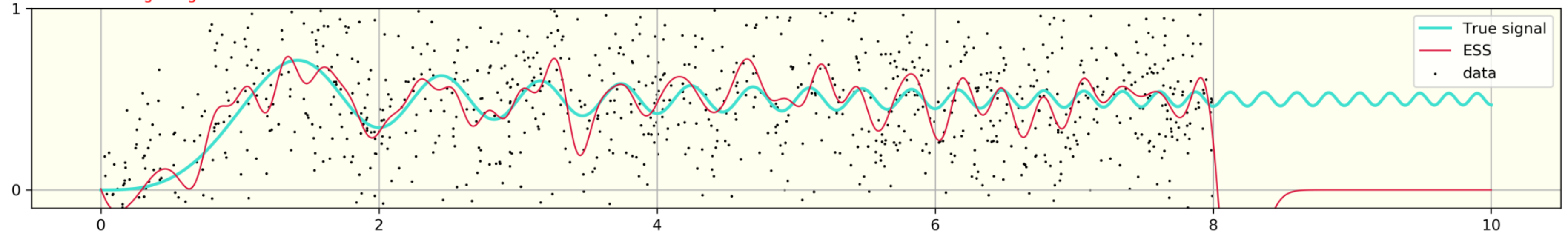
Laplacian Kernel Ridge Regression fitted to the Fresnel S function



Sigmoid Kernel Ridge Regression fitted to the Fresnel S function

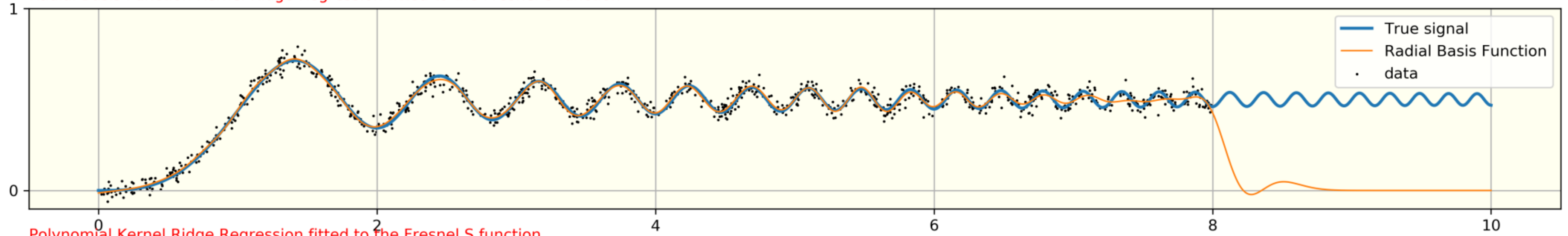


ESS Kernel Ridge Regression fitted to the Fresnel S function

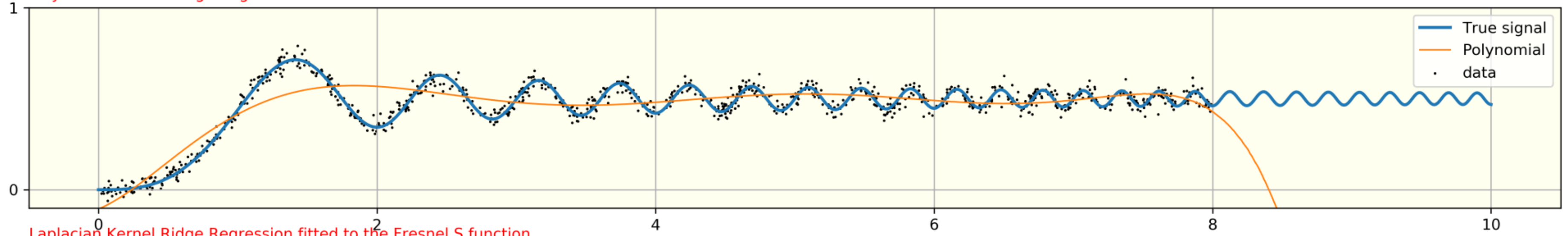


This slide and the next one compare the Fresnel S function with high noise, SNR = 1.3 (this slide) the next slide is the same function with SNR = 2.9

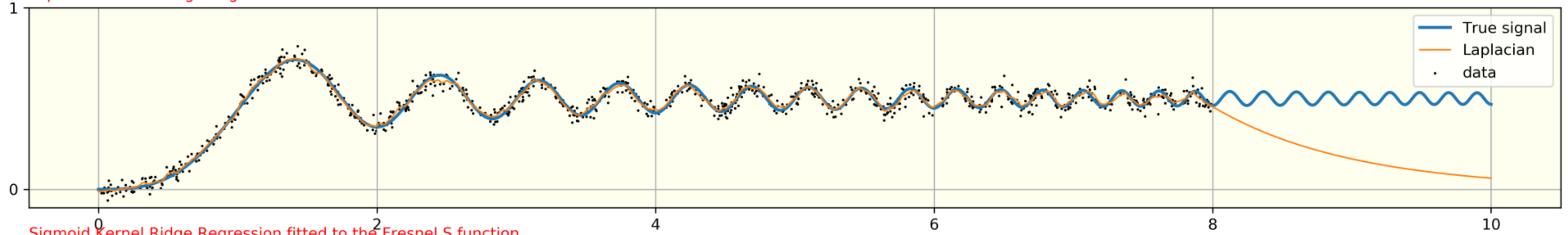
Radial Basis Function Kernel Ridge Regression fitted to the Fresnel S function



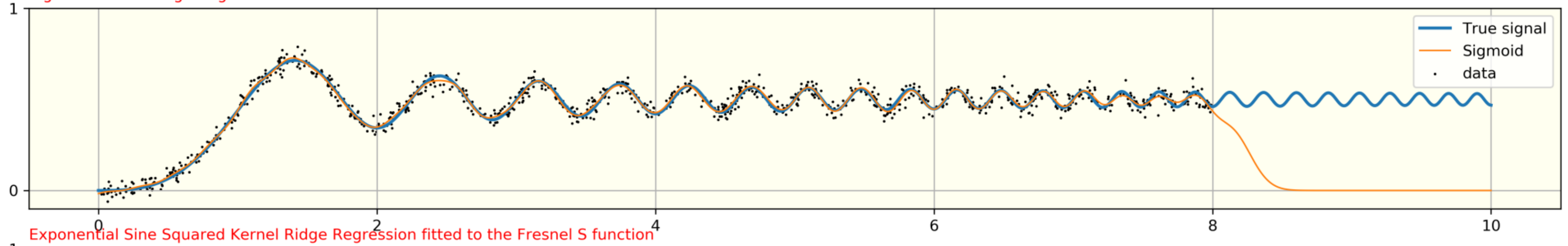
Polynomial Kernel Ridge Regression fitted to the Fresnel S function



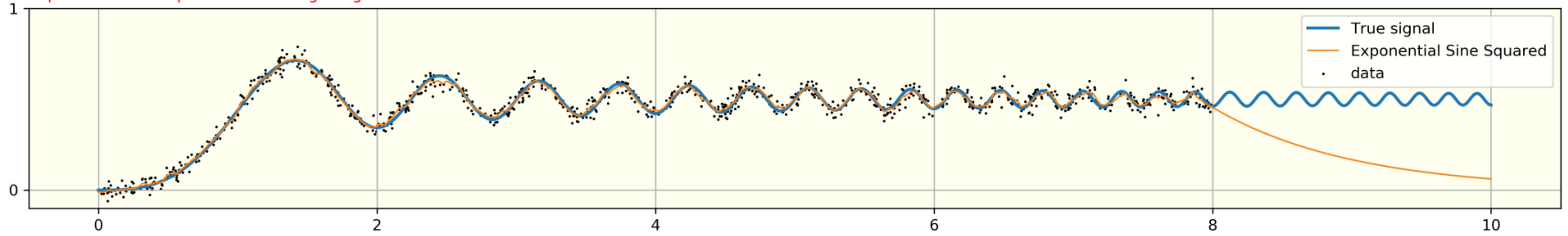
Laplacian Kernel Ridge Regression fitted to the Fresnel S function



Sigmoid Kernel Ridge Regression fitted to the Fresnel S function



Exponential Sine Squared Kernel Ridge Regression fitted to the Fresnel S function



Discussion

Since the usual ridge regression can’t handle nonlinearities in the data, the kernel trick is used to overcome its deficiencies, it is also more suitable for higher dimensional data however this feature is not analyzed here.

Kernel based methods, are instance learners; this means that the kernel does not learn a set of parameters that appertain to the model rather it defines weights for each data point that lower the error score. So this explains why most of the kernels are incapable of prediction out of the data range. Although some are better at generalizing nevertheless which will be further discussed.

Except for the RBF kernel the parameters seem to have not have much of an influence of the behavior of the regression. I tested this for the Fresnel C function with lower SNR only and saw no difference for the range of parameters I tested (which lied in a reasonable range) and decided not to show the plot.

Discussion(continued)

- The radial basis function seems to be very robust over a range of shapes and noise amounts. It is based on the squared Euclidean distance between the data points and a fixed point. Since we are not using highly noisy data it performs reasonably well. Moreover, it’s a kernel of choice for predicting time series, which have more or less sinusoidal shapes and since we are mainly using such functions it is relevant here. My guess for the generalization capabilities of the RBF kernel is that it tries to fit a function to the parameters based on the weights it finds. For example, if the weights found in the vicinity of the last data points are negative (denoting a decreasing domain in the function) the fitted line will decrease based on the weights (also denoting the slope) until at some point it reaches zero as there are no more data points and all weights are assigned to zero.
- The polynomial Kernel is in general not suitable to apply to repeating functions as they tend to have many zeroes and polynomials cannot capture their shapes very well. Moreover, this type of kernel suffers from numerical instability. For example in the case of noisy Fresnel S function we see that the end of the regression curve is highly unstable. As the absolute value of the polynomial kernel becomes larger than 1 it starts to tend to infinity and this could a culprit behind the unusual behavior of the polynomial kernel in this particular case.
- The Laplacian kernel is a variant of the RBF kernel using Manhattan distance. It is more suitable for noiseless data. My guess for its general worse performance is the exponentiation used in this kernel and makes it more sensitive to how the data are distributed. The generalization here is bad and the weights exponentially (albeit slowly) decay towards zero.
- The sigmoid Kernel is equivalent to a two layer perceptron and as such it can find nonlinearities in the data pretty well, but nevertheless has worse performance than exponential sine squared, RBF or laplacian kernels and is less robust against noise.
- The exponential sine squared kernel was specifically designed to capture repeating functions such as the trigonometric functions it is parametrized by a length scale and periodicity which could very well model repeating functions. For trigonometric functions this kernel has best generalization capabilities as it is defined based on how a sinusoidal function would behave finding optimal periodicity and wavelength.