PRINCIPLES OF MACHINE LEARNING

Exercise sheet 5 - regularization and the kernel trick

11.01.2024

Richard Restetzki	2973740
Akmalkhon Khashimov	50178353
Valdrin Smakaj	50138041
Suraj Giri	50190564
Nijat Sadikhov	50186266
Nikita Morev	50134788
Aleksandr Semenikhin	50118777
Muslimbek Abduvaliev	50136555
Suyash Thapa	50205756
Mohammad Mehdi Deylamipour	50009389
Nicolás López Funes	50151598
Andrii Shevliakov	50198078
Nurmukhammad Aberkulov	50102727

5.1.1: regularized least squares

```
def phi(x, d):
    return np.vander([x], d+1, increasing=True).flatten()
```

Parameters:

x (float): The input value.

d (int): The degree of the polynomial.

Returns:

numpy.ndarray: The feature map vector.

Increase in the number of iterations results in increase of proximity/correctness of the estimated mean.

We use np.vander with increasing=False to create the feature map, such that the powers are in the order x^0, x^1, ..., x^d.

We only need the first column (x^0 to x^d), hence `[:, :d+1]`.

5.1.2: Phi vectorized

```
def phi_vectorized(x, d):
    return np.vander(x, d+1, increasing=True)
```

Parameters:

  x (float): The input value.

  d (int): The degree of the polynomial.

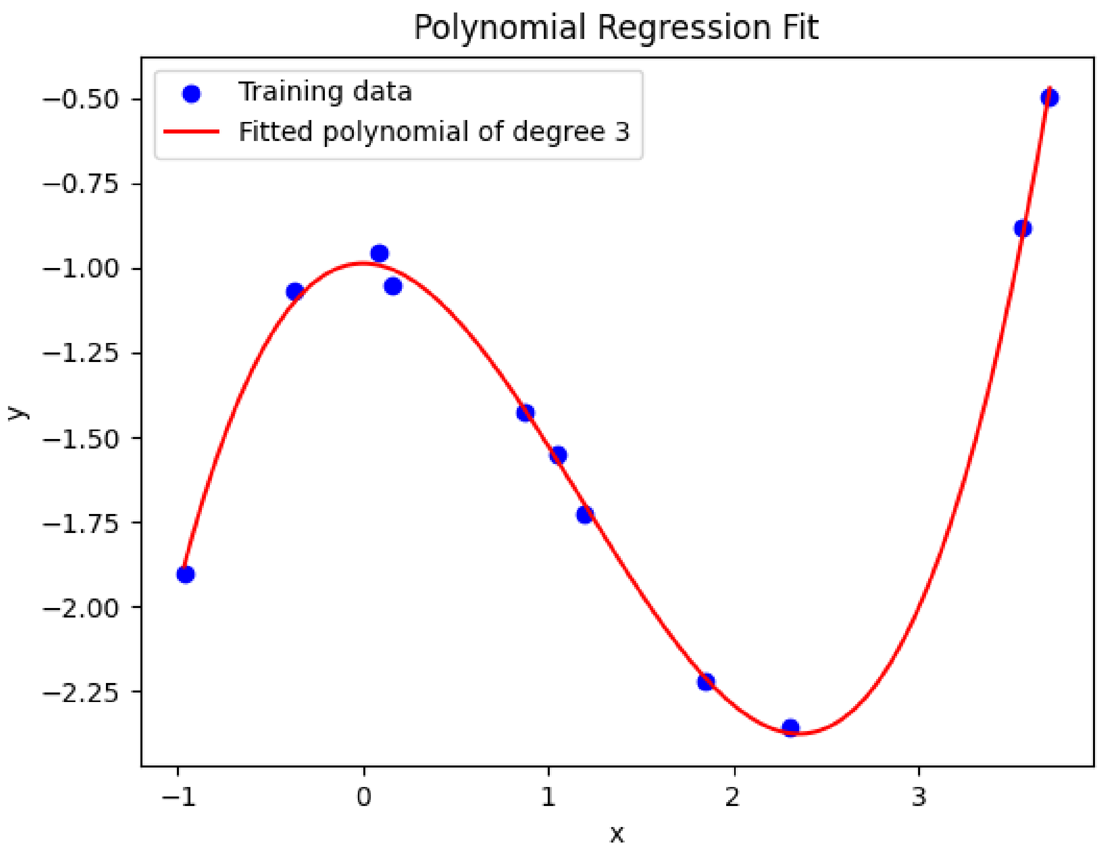
Returns:

  numpy.ndarray: The feature map vector.

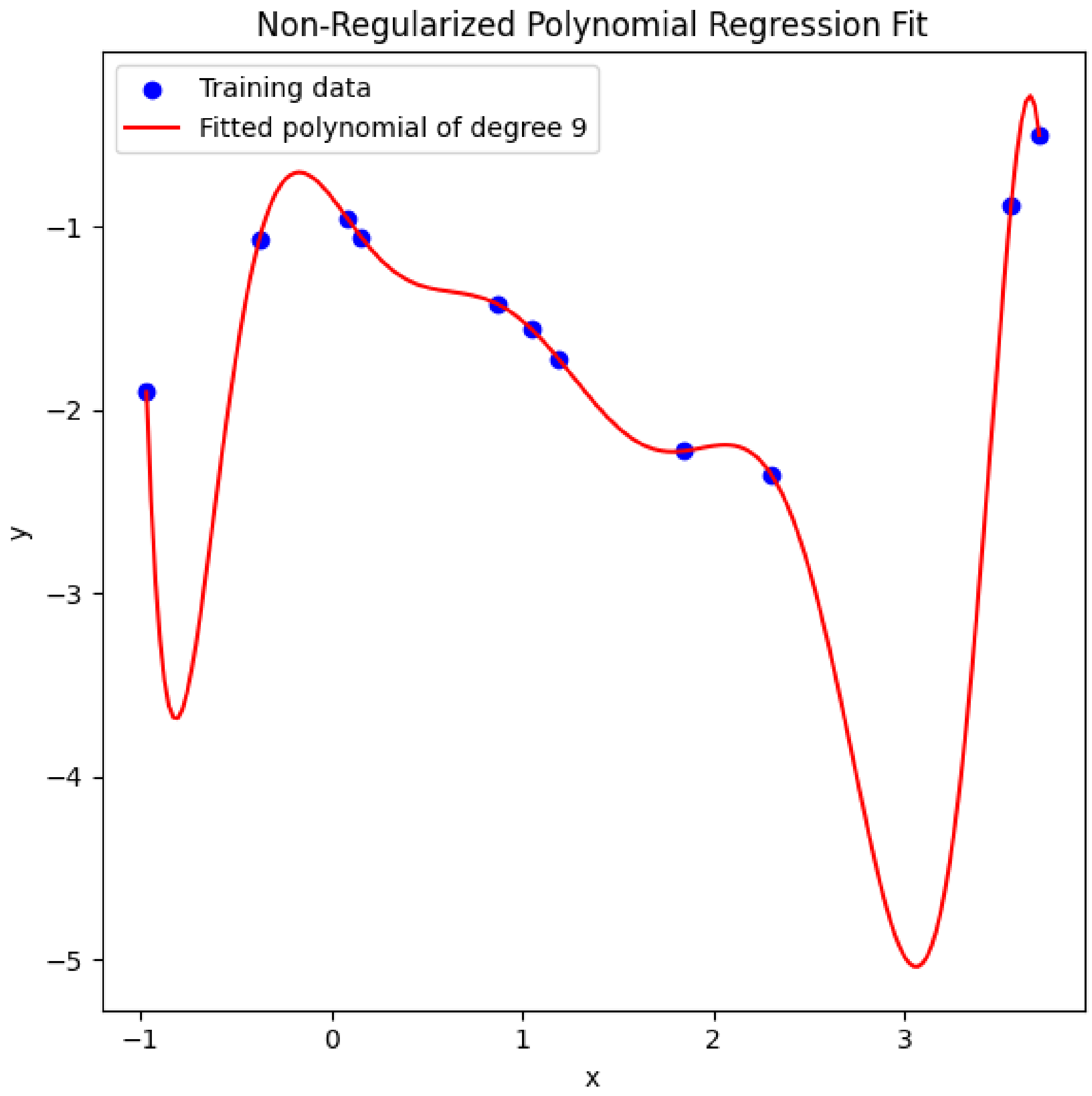
Function to create a feature map for polynomial regression for a vector of input values

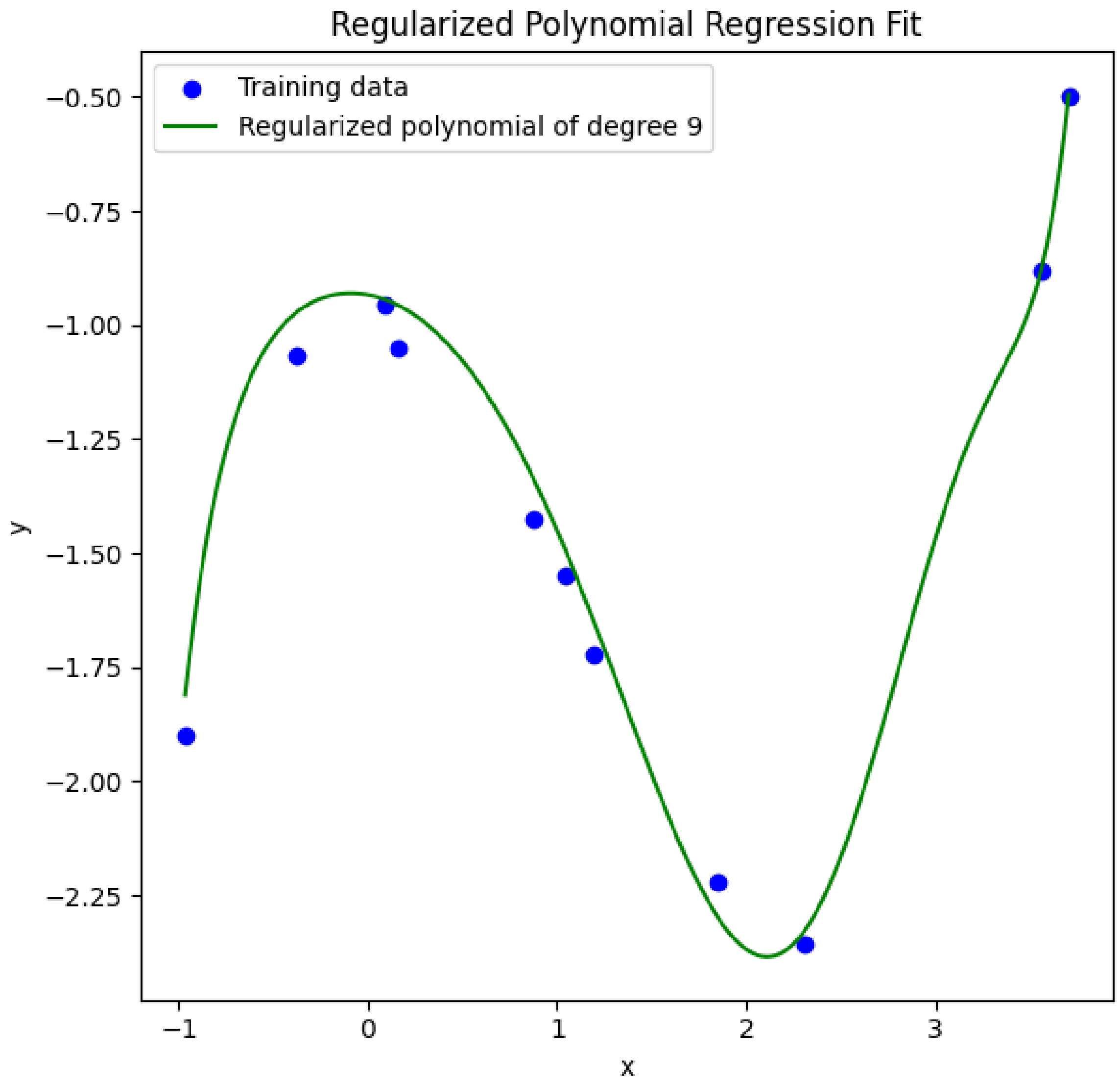
The input x is now a vector, so we create a feature matrix,

where each row is the feature map for a single value of x.



5.1.3: result





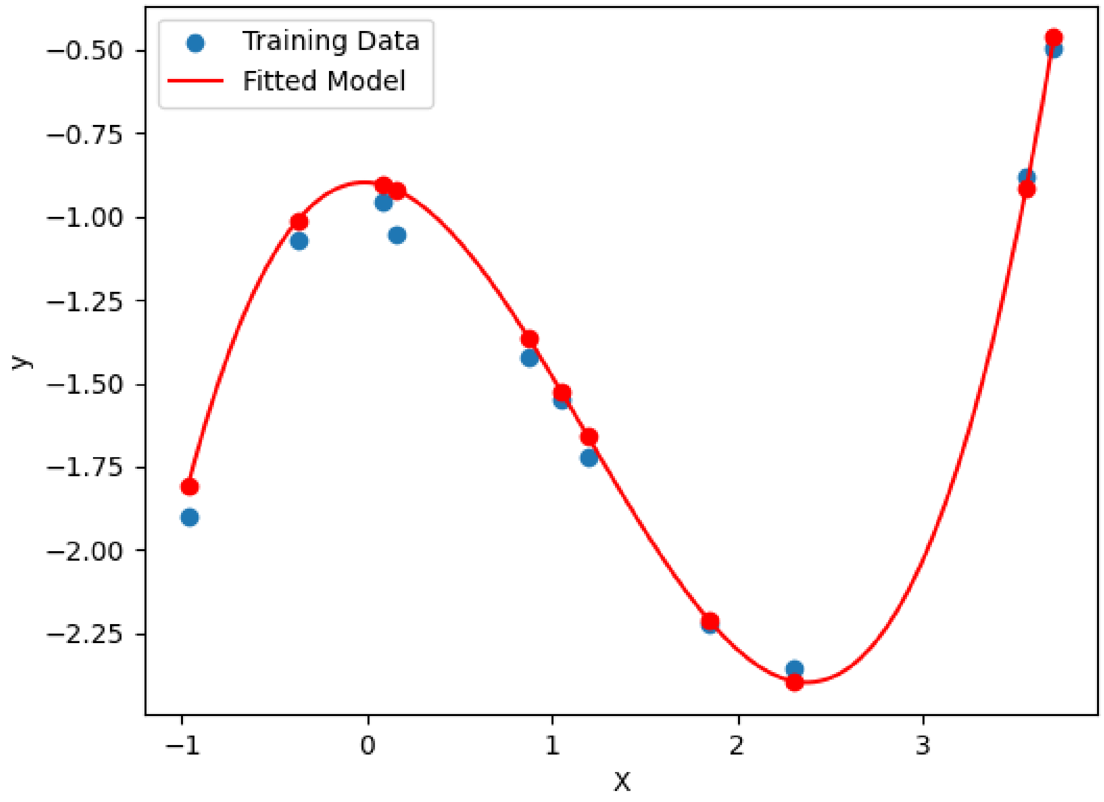
Which model would you say gives a “subjectively” more reasonable description of the data?
Regularized – it has no dips compared to non-regularized.

5.2: kernel least squares

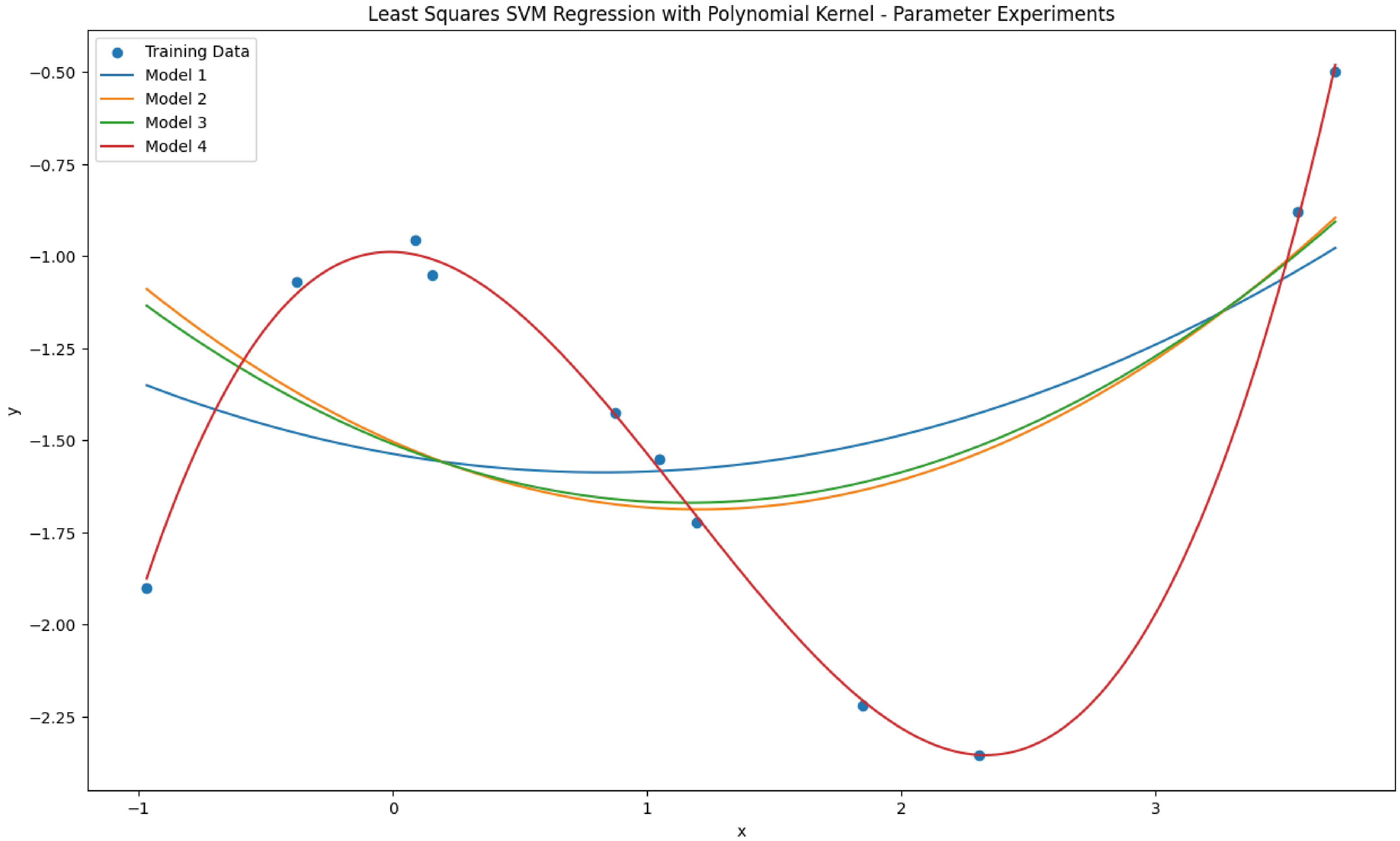
The fitted model describes the training data pretty accurately. But even a slight change in the model parameters leads to unsatisfying results.

Can you recognize a connection between kernel least squares and Gaussian processes?

The connection between Kernel Least Squares and Gaussian Processes lies in the shared use of kernel functions to model relationships or similarities between data points.



5.3: least squares SVMs for regression



5.3: least squares SVMs for regression

Effect of C:

Smaller values of C lead to a smoother fit, emphasizing the importance of a simple model.

Larger values of C result in a more complex model, potentially fitting the training data more closely but at the risk of overfitting.

Effect of b:

Lower values of b tend to compress the influence of individual data points, resulting in a more globally shaped model.

Higher values of b may amplify the impact of individual data points, leading to a more locally responsive model.

Effect of d:

Increasing the polynomial degree (d) allows the model to capture more intricate patterns in the data.

However, excessively high values of d might cause the model to fit the noise in the data, leading to overfitting.

Overall:

The choice of parameters influences the trade-off between model simplicity and data fitting.

Experimenting with different parameter sets helps to understand the flexibility and sensitivity of the model to changes in C, b, and d.

It's essential to strike a balance between fitting the training data well and generalizing to unseen data, avoiding both underfitting and overfitting.

5.4: kernel SVMs for binary classification

We could not achieve the solution, but our progress in a file attached.

5.5: kernel minimum enclosing balls

We could not achieve the solution, but our progress in a file attached.