9. Cubic Features

Project due Mar 15, 2023 08:59 -03 Past due

In this section, we will work with a **cubic feature** mapping which maps an input vector x new feature vector $\phi(x)$, defined so that for any $x, x' \in \mathbb{R}^d$:

$$\phi(x)^T \phi(x') = (x^T x' + 1)^3$$

You will be working in the files `part1/main.py` and `part1/features.py` in this prob

Computing Cubic Features

0.0/3.0 points (graded)

In 2-D, let $x = [x_1, x_2]$. Write down the explicit cubic feature mapping $\phi(x)$ as a vect $\phi(x) = [f_1(x_1, x_2), \cdots, f_N(x_1, x_2)]$

? STANDARD NOTATION

Hint

$\phi(x) =$

Submit You have used 0 of 20 attempts

The `cubic_features` function in `features.py` is already implemented for you. That f with an arbitrary dimension and compute the corresponding features for the cubic Kern don't leverage the kernel properties that allow us to do a more efficient computation wi (without computing the features themselves). Instead, here we do compute the cubic f apply the PCA on the output features.

Applying to MNIST

1.0/1.0 point (graded)

If we explicitly apply the cubic feature mapping to the original 784-dimensional raw pix representation would be of massive dimensionality. Instead, we will apply the cubic fea dimensional PCA representation of our training data which we will have to calculate just

Submit You have used 1 of 20 attempts

Polynomial svm using scikit-learn

0.0/1.0 point (graded)

If we explicitly apply the cubic polynomial svm to the original 784-dimensional raw pixe representation would be of massive dimensionality. Instead, we will apply the cubic pol dimensional PCA representation of our training data which we will have to calculate just 18-dimensional representation in the previous problem. Use the sklearn package and b your local machine. Use random_state = 0, kernel = 'poly', degree = 3, and default value

If you have done everything correctly, cubic polynomial svm should perform better (on features than either the 18-dimensional principal components or raw pixels. The error c cubic polynomial svm should only be around 0.06, demonstrating the power of nonlinea

Error rate for 10-dimensional PCA features using cubic polynomial svm =

Submit You have used 0 of 20 attempts

Rbf svm using scikit-learn

0.0/1.0 point (graded)

We will apply the rbf svm to the 10-dimensional PCA representation of our training data calculate just as we calculated the 18-dimensional representation in the previous proble package and build the SVM model on your local machine. Use random_state = 0, kerne values for other parameters.

If you have done everything correctly, rbf svm should perform better (on the test set) u either the 18-dimensional principal components or raw pixels. The error on the test set should only be around 0.05, demonstrating the power of nonlinear classification models

Error rate for 10-dimensional PCA features using rbf svm =

Submit You have used 0 of 20 attempts

Discussion

Hint for Polynomial svm using scikit-learn and Rbf svm using scikit-learn
Feed in train_pca10 instead of cubic transformed features, the sklearn SVC function will do the cubic transfo

Inuitively, why do we use cubic features?
And how is it possible that this can be more accurate than the original data?

Applying to MNIST Parameters
I am currently doing softmax_regression on the cubic feature mapping to the train_pca10 set. I am using tem

unable to write the indexes anfd the notation is not explained in the guide for Computing Cub
unable to write the indexes anfd the notation is not explained in the guide for Computing Cubic Features I us

Applying to MNIST - confused [SOLVED]
Hi, I simply may be doing something wrong, but I'm hoping for some clarification. I'm working on the "Applyin

Polynomial svm using scikit-learn
This is wild, I used get_MNIST_data for my data input to the train_x,train_y,test_x. And I mistakenly applied Rb

PCA 10 and cubic function test error
I used Pca to reduce to 10 and then applied the given cubic function and then did the softmax function with

Polynomial/Rbf svm using scikit-learn - Is there a bug? [SOLVED]
It looks like to find a correct test value for the poly kernel we need to use the rbf kernel. Unfortunately, it doe