9. Cubic Features

🔖 Bookmark this page

Project due Oct 21, 2020 05:29 IST *Completed*

In this section, we will work with a **cubic feature** mapping which maps an input vector $x = [x_1, \ldots, x_d]$ into a 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 problem

Computing Cubic Features

3.0/3.0 points (graded)

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

> STANDARD NOTATION

> **Hint**
>
> Show

$\phi(x) = \boxed{\text{[x\_1^3,x\_2^3,sqrt(3)*x\_1^2*x\_2,sqrt(3)*x\_1*x\_2^2,sqrt(3)}}$ ✔

Answer: [x_1^3, sqrt(3)*x_1^2*x_2, sqrt(3)*x_1^2, sqrt(3)*x_1*x_2^2, sqrt(6)*x_1*x_2, sqrt(3)*x_1, x_2^3, sqrt(3)*x_2^2, sqrt(3)*x_2, 1]

> Submit   You have used 2 of 20 attempts

ℹ   Answers are displayed within the problem

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

Applying to MNIST

0.0/1.0 point (graded)

If we explicitly apply the cubic feature mapping to the original 784-dimensional raw pixel features, the resulting representation would be of massive dimensionality. Instead, we will apply the cubic feature mapping to the 10-dimensional PCA representation of our training data which we will have to calculate just as we calculated the 18-dimensional representation in the previous problem. After applying the cubic feature mapping to the PCA representations for both the train and test datasets, retrain the softmax regression model using these new features and report the resulting test set error below.

Important: You will probably get a runtime warning for getting the log of 0, ignore. Your code should still run and perform correctly.

Note: Use the same training parameters as the first softmax model given in `main.py` file and temperature 1.

If you have done everything correctly, softmax regression should perform better (on the test set) using these features than either the 18-dimensional principal components or raw pixels. The error on the test set using cubic features should only be around 0.08, demonstrating the power of nonlinear classification models.

features should only be around 0.08, demonstrating the power of nonlinear classification models.

Error rate for 10-dimensional cubic PCA features = | 0.0865 | ✖ **Answer:** 0.0849

| Submit | You have used 4 of 20 attempts |

ℹ Answers are displayed within the problem

Discussion

Topic: Unit 2 Nonlinear Classification, Linear regression, Collaborative Filtering (2 weeks):Project 2: Digit recognition (Part 1) / 9. Cubic Features

Add a Post

| Show all posts ⌄ | by recent activity ⌄ |

?	[STAFF] Applying To MNIST-The grader is not accepting the answer, although it is near the specified value	5
?	Where can I find the solution code I could not get the correct error rate for 10-dim cubic PCA features in spite of implementing the steps as was discussed in some othe...	8
💬	[Staff] Clarity re the computing cubic features- I might be the only person who misunderstood this question, but may I ask that the question is clar...	2
☑	Kernel Confusion - How are they used? After watching a few youtube videos and rewatching the lectures, I think I understand what a kernel does and why it is such an effici...	6
💬	10 dimensional cubic pca error I have constructed 10-dimensional CA features, then apply cubic feature mapping to train_pca10 and test_pca10. I got error around ...	27
?	1.5/3.0 points on 10-dim vector I entered my ten terms, based on the laborious algebra. I was initially surprised to get half credit; now I have double checked my res...	3
💬	Time taken in softmax_regression call It is very interesting that the time taken in the softmax_regression call does not vary much between raw, pca and cubic cases. For m...	6
💬	Help on Computing Cubic Features I got partial credit(.0.9) by expanding (_x1+x2)+1)^3, can someone point to correct link to understand it?	8
?	problem with error rate 10 dimensional PCA My softmax_regression has passed the tests and worked well until this part. I feed in the test_cube which is proper shape (60000, 28...	4
💬	x' using standard notation how do i write (1+x1*x'1+x2*x2') in equation i tried (1+x_1*x'_1+x_2*x'_2) but it did not work got parsing error	3
?	Keep getting [Math Processing Error] the moment I start using raise to ^ symbol I get [Math Processing Error]	3
☑	Applying to MNIST - "quadratic"? To me, the word "quadratic" is not consistent with the rest of the exercise's text. Did the author actually mean to write "cubic"?	3
💬	Problem 9: x is a column vector, right? the feature vector expression expects a column vector, but the problem definition does not state that it is a column vector and in th...	2

| ‹ Previous | Next › |

edX

About

Affiliates

edX for Business

Open edX

Careers

News

Legal

Terms of Service & Honor Code

Privacy Policy

Accessibility Policy

Trademark Policy

Sitemap

Connect

Blog

Contact Us

Help Center

Media Kit

Donate