



MITx 6.86x

Machine Learning with Python-From Linear Models to Deep Learning

[Help](#)

smitha_kannur ▾

[Course](#)

[Progress](#)

[Dates](#)

[Discussion](#)

[Resources](#)

[Home](#) / [Course](#) / [Unit 2 Nonlinear Classification, Linear regression, Collaborative Filtering \(2 weeks\)](#) / [Project 2: Digit recognition \(Part 1\)](#)

[< Previous](#)



[Next >](#)

10. Kernel Methods

[Bookmark this page](#)

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

As you can see, implementing a direct mapping to the high-dimensional features is a lot of work (imagine using an even higher dimensional feature mapping.) This is where the kernel trick becomes useful.

Recall the kernel perceptron algorithm we learned in the lecture. The weights θ can be represented by a linear combination of features:

$$\theta = \sum_{i=1}^n \alpha^{(i)} y^{(i)} \phi(x^{(i)})$$

In the softmax regression fomulation, we can also apply this representation of the weights:

$$\theta_j = \sum_{i=1}^n \alpha_j^{(i)} y^{(i)} \phi(x^{(i)}) .$$

$$h(x) = \frac{1}{\sum_{j=1}^k e^{[\theta_j \cdot \phi(x)/\tau] - c}} \begin{bmatrix} e^{[\theta_1 \cdot \phi(x)/\tau] - c} \\ e^{[\theta_2 \cdot \phi(x)/\tau] - c} \\ \vdots \\ e^{[\theta_k \cdot \phi(x)/\tau] - c} \end{bmatrix}$$

$$h(x) = \frac{1}{\sum_{j=1}^k e^{[\sum_{i=1}^n \alpha_j^{(i)} y^{(i)} \phi(x^{(i)}) \cdot \phi(x)/\tau] - c}} \begin{bmatrix} e^{[\sum_{i=1}^n \alpha_1^{(i)} y^{(i)} \phi(x^{(i)}) \cdot \phi(x)/\tau] - c} \\ e^{[\sum_{i=1}^n \alpha_2^{(i)} y^{(i)} \phi(x^{(i)}) \cdot \phi(x)/\tau] - c} \\ \vdots \\ e^{[\sum_{i=1}^n \alpha_k^{(i)} y^{(i)} \phi(x^{(i)}) \cdot \phi(x)/\tau] - c} \end{bmatrix}$$

We actually do not need the real mapping $\phi(x)$, but the inner product between two features after mapping: $\phi(x_i) \cdot \phi(x)$, where x_i is a point in the training set and x is the new data point for which we want to compute the probability. If we can create a kernel function $K(x, y) = \phi(x) \cdot \phi(y)$, for any two points x and y , we can then kernelize our softmax regression algorithm.

You will be working in the files `part1/main.py` and `part1/kernel.py` in this problem

Implementing Polynomial Kernel

1.0/1 point (graded)

In the last section, we explicitly created a cubic feature mapping. Now, suppose we want to map the features into d dimensional polynomial space,

$$\phi(x) = \langle x_d^2, \dots, x_1^2, \sqrt{2}x_d x_{d-1}, \dots, \sqrt{2}x_d x_1, \sqrt{2}x_{d-1} x_{d-2}, \dots, \sqrt{2}x_{d-1} x_1, \dots, \sqrt{2}x_2 x_1, \sqrt{2}c x_d, \dots, \sqrt{2}c x_1, c \rangle$$

Write a function `polynomial_kernel` that takes in two matrix X and Y and computes the polynomial kernel $K(x, y)$ for every pair of rows x in X and y in Y .

Available Functions: You have access to the NumPy python library as `np`

```
1 def polynomial_kernel(X, Y, c, p):
2     """
```

```

3      Compute the polynomial kernel between two matrices X and Y::
4      K(x, y) = (<x, y> + c)^p
5      for each pair of rows x in X and y in Y.
6
7      Args:
8          X - (n, d) NumPy array (n datapoints each with d features)
9          Y - (m, d) NumPy array (m datapoints each with d features)
10         c - a coefficient to trade off high-order and low-order terms (scalar)
11         p - the degree of the polynomial kernel
12
13     Returns:
14         kernel_matrix - (n, m) Numpy array containing the kernel matrix
15     """
16     # YOUR CODE HERE

```

Press ESC then TAB or click outside of the code editor to exit

Correct

Test results

CORRECT

[See full output](#)

[See full output](#)

Submit

You have used 2 of 25 attempts

Gaussian RBF Kernel

1.0/1 point (graded)

Another commonly used kernel is the Gaussian RBF kernel. Similarly, write a function `rbf_kernel` that takes in two matrices X and Y and computes the RBF kernel $K(x, y)$ for every pair of rows x in X and y in Y .

Available Functions: You have access to the NumPy python library as np

```

1 def rbf_kernel(X, Y, gamma):
2     """
3     Compute the Gaussian RBF kernel between two matrices X and Y::
4     K(x, y) = exp(-gamma ||x-y||^2)
5     for each pair of rows x in X and y in Y.
6
7     Args:
8         X - (n, d) NumPy array (n datapoints each with d features)
9         Y - (m, d) NumPy array (m datapoints each with d features)
10        gamma - the gamma parameter of gaussian function (scalar)
11
12    Returns:
13        kernel_matrix - (n, m) Numpy array containing the kernel matrix
14    """
15    # YOUR CODE HERE
16    try:

```

Press ESC then TAB or click outside of the code editor to exit

Correct

Test results

CORRECT

[See full output](#)

[See full output](#)

Submit

You have used 1 of 25 attempts

Now try implementing the softmax regression using kernelized features. You will have to rewrite the

Now, try implementing the softmax regression using kernelized features. You will have to rewrite the `softmax_regression` function in `softmax.py`, as well as the auxiliary functions `compute_cost_function`, `compute_probabilities`, `run_gradient_descent_iteration`.

How does the test error change?

In this project, you have been familiarized with the MNIST dataset for digit recognition, a popular task in computer vision.

You have implemented a linear regression which turned out to be inadequate for this task. You have also learned how to use scikit-learn's SVM for binary classification and multiclass classification.

Then, you have implemented your own softmax regression using gradient descent.

Finally, you experimented with different hyperparameters, different labels and different features, including kernelized features.

In the next project, you will apply neural networks to this task.

Discussion

Hide Discussion

Topic: Unit 2 Nonlinear Classification, Linear regression, Collaborative Filtering (2 weeks):Project 2: Digit recognition (Part 1) / 10. Kernel Methods

Add a Post

Show all posts	by recent activity
<p>? Rewriting Softmax Hello! Can anyone help with rewriting Softmax with kernel method? Basically I am unsure how to update the Gradient in Gradient D...</p>	1
<p>🗨️ staff's solution for rewriting functions to run softmax regression using kernelized features Is there any staff's solution for rewriting functions to run softmax regression using kernelized features? If so please update, thanks.</p>	1
<p>🗨️ What if one of the labels is 0?</p>	3
<p>🗨️ [staff] RBF Would be really nice to know what I'm missing I can't seem to figure it out. I calculated the norm of X, the norm of Y, then I did the distance of those two. In this logic, What is wron...</p>	11
<p>? [staff] Grader fails My code passes with the checks on test.py but it fails when submitting it. As far as I can tell the output of my implementation seems...</p>	27
<p>? Help with rewriting softmax? I'm done with all graded parts of the project and got them all correct. But I'm having trouble with the final challenge at the end about...</p>	3
<p>🗨️ Thank You Thank you so much, it was a great experience. Even though the logistic regression along with the PCA was not covered in the lecture...</p>	1
<p>🗨️ For loops I created my Kernel functions using simple for loops, just a couple of lines of code. However, I feel like there should be a smarter wa...</p>	7
<p>🗨️ Gaussian RBF Kernel - Stuck Hi, I'm stuck at the Gaussian RBF Kernel part. It doesn't seem very complicated but i dont get it.. Any advice ? May be i misunderstood...</p>	13
<p>🗨️ Polynomial Kernel Dont we have a testing method in test.py or main.py to test this code locally before submitting ?</p>	12
<p>✅ How to get ML intuition and truly understand the projects? How do you get intuition and learn from this project (and the course, generally)? On my side, I'm doing all individual steps smoothly,...</p>	18
<p>? Now, try implementing the softmax regression using kernelized features</p>	9
<p>🗨️ Output matrix dimension The problem states that the output matrix (i.e. kernel matrix) has dimension (n x m). I do not understand why m columns. x-y is...</p>	3
<p>? Kernel Matrices</p>	

< Previous

Next >

© All Rights Reserved



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](#)



© 2020 edX Inc. All rights reserved.

深圳市恒宇博科技有限公司 [粤ICP备17044299号-2](#)