Multivariable Parametric Regression

# Problem Statement

The multivariable feature data set should be mapped to a higher dimension. The data should be trained and tested using the Linear Regression model, regression through iterative solution and Gaussian kernel function. Based on the fit, the labels have to be predicted and the predicted labels are compared to the actual values to measure the accuracy of the model. The fit represents finding the coefficient of the feature computed with training data which is then used to predict the unknown labels from the test data. The 10-fold cross validation has to be done to analyze the performance of the model. The accuracy of the model is analyzed with the training and testing error calculated by Mean Square Error.

# Problem Solution

The coefficient of features is calculated by Linear Regression model, regression through iterative solution and Gaussian kernel function Regression.

Linear Regression is applied to different mapping of the dataset. The Linear Regression used in Single Variable is used here too.

For iterative solution gradient descent algorithm is implemented and the dual solution is solved Gaussian radial basis function kernel, or RBF kernel.

# Implementation Details

I have implemented the code in ipython notebook. The filename has to be mentioned and in the tool bar option “Cell” -> “Run All” will implemented the whole file and the results will be printed.

# Results and Discussion

1. **Mapping to higher dimension**: file "mvar-set1.dat.txt"

Data set mapped to higher dimension space of degree 3

Data Matrix mapped to Higher dimemsion of degree 3

[[ 1.00000000e+00 1.67346939e+00 4.48979592e-01 ..., 1.25736725e+00

3.37342434e-01 9.05065066e-02]

[ 1.00000000e+00 -4.08163265e-02 5.30612245e-01 ..., 8.83985414e-04

-1.14918104e-02 1.49393535e-01]

[ 1.00000000e+00 -7.75510204e-01 -1.59183673e+00 ..., -9.57356204e-01

-1.96509958e+00 -4.03362545e+00]

...,

[ 1.00000000e+00 -2.00000000e+00 -6.93877551e-01 ..., -2.77551020e+00

-9.62932112e-01 -3.34078488e-01]

[ 1.00000000e+00 -1.83673469e+00 1.02040816e+00 ..., 3.44244320e+00

-1.91246844e+00 1.06248247e+00]

[ 1.00000000e+00 -2.85714286e-01 1.42857143e+00 ..., 1.16618076e-01

-5.83090379e-01 2.91545190e+00]]

The data set has been mapped to five different degrees 2,3,4,5,6.

1. **Performing Linear Regression on the Mapped Data set**

Predicted and actual values of high dimensional data set of degree 3(first 25 values)

Predcited Actual

[ 3.07736342] [2.861980967575097]

[ 1.51365053] [0.5195252261416367]

[-1.38696723] [-2.245760962347731]

[-2.32471805] [-1.561989674513609]

[ 1.34557589] [1.887592936186108]

[ 1.22770058] [0.9884311316371547]

[-1.44192419] [-1.085317769937612]

[ 1.29287274] [0.8038785273144268]

[ 2.49595503] [3.18204127016383]

[ 2.53802138] [2.351083268524211]

[ 3.51292129] [3.626071229401944]

[-0.17188473] [0.4634078745339167]

[ 0.85542858] [1.487055672161672]

[ 2.3113871] [2.142605152020115]

[-0.46229492] [-1.110575348573327]

[ 3.10202926] [3.929068438319104]

[-1.94229288] [-1.927879404879771]

[-0.56049346] [-0.1816051916186905]

[ 1.90961133] [1.71526336810711]

[ 0.62064496] [0.7650547285963669]

[-0.87883622] [-1.280690811878441]

[ 3.69919814] [3.876430712322037]

[ 0.59320767] [0.5123972650771516]

[ 0.42276492] [0.3184214808888129]

[ 2.76858808] [2.495057994255536]

**The coefficients of higher dimension data set**

Theta value of high dimensional data set of degree 3

[[ 1.02230218e+00]

[ 1.00248539e+00]

[ 1.00537442e+00]

[ -1.26097513e-02]

[ -8.47393334e-03]

[ -6.43213699e-03]

[ -2.31159943e-03]

[ -1.64092663e-02]

[ 5.84897669e-04]

[ 3.15715316e-03]]

**The Mean square Test Error of all high dimension data set**

Mean Square Errors of High Dimensional feature space

Degree MSE

2 0.258256248391

3 0.257655351948

4 0.256493824893

5 0.256237649361

6 0.255132612013

**Cross validation on high dimension feature space**

The Training Errot and Testing Error for Linear Regression on High Dimensional Feature Space of degree 6 for 10 folds

Training Error Testing Error

1.721218 2.025014

5.277094 2.139538

6.931781 2.052602

7.224948 2.054735

8.520127 2.129154

7.227964 1.953868

7.575019 2.151559

7.194835 2.074939

8.557931 2.150347

7.581050 2.147773

1. **Explicit Vs Iterative Solution**

**Iterative Solution**

The gradient descent algorithm is used to solve regression through iterative solution. The training set is the first 2000 records of the data and the remaining is used for testing.

Learning rate = 0.005

Iteration = 100

Theta\_zero = 0.1

**Theta**

[ -1.59841357e+212 -1.07735597e+213 -3.28323529e+213 -3.54231269e+212

-9.17404054e+211 -3.66557925e+212 -2.99823948e+213 -5.22636093e+213

-1.72158510e+213 -9.24487721e+213]

Learning rate = 0.00005

Iteration = 600

Theta\_zero = 0.1

**Theta**

[ 1.01512903 1.00395836 0.98701184 -0.01487465 -0.01156393 -0.00260941

-0.00413616 -0.01150208 0.00471191 0.00403633]

**Explicit Solution**

**Theta**

[[ 1.01512825]

[ 1.0039924 ]

[ 0.98704724]

[-0.01487432]

[-0.01156381]

[-0.00260912]

[-0.0041459 ]

[-0.01150636]

[ 0.00470772]

[ 0.00402614]]

1. **Gaussian Kernel Function**

Kernel method was performed on higher dimension feature space with degree 2.

**Alpha**

[[ 2.39499829]

[-0.61700647]

[-2.10648385]

[-1.55336294]

[ 0.94903295]

[ 0.97338585]

[-0.96511813]

[ 0.80051309]

[ 2.99515497]

[ 1.82511589]

[ 3.48163753]

[ 0.48699148]

[ 0.80591897]

[ 1.97621425]

[-1.00537791]

[ 3.6091703 ]

[-1.91318885]

[ 0.14868346]

[ 0.46777488]

[ 0.76375698]

[-1.17075398]

[ 3.8133481 ]

[ 0.28796778]

[ 0.32171543]

[ 2.12579566]

[ 1.57983271]

[ 1.20248007]

[ 0.17015024]

[-0.30118843]

[ 2.13755283]

[-1.5336172 ]

[ 0.75463089]

[ 1.4996998 ]

[ 1.34419739]

[-0.81658818]

[-0.23977971]

[ 2.04633209]

[ 1.32937515]

[ 0.71692488]

[ 0.64235499]

[-0.02211506]

[-1.16172165]

[-0.39254188]

[-1.13943073]

[-1.11250928]

[-0.99161622]

[-1.64664593]

[ 0.84908324]

[-2.10676256]

[-1.21048872]]

**Predicted based on alpha**

[[ 2.86198097 0.51952523 -2.24576096 ..., -0.66468951 -0.10755482

2.20162969]]

**Mean Square Error**

Mean Error

[ 4.90950493e-27]

**Gradient Descent Vs Gaussian Kernel**

The number of iterations in the gradient descent is responsible for the time take for the function to find theta. If the iteration is small the theta values are large which makes the accuracy too low and if the iteration is large then the time taken is more.

Hence Gaussian Kernel is better in performance for the higher dimension feature space of degree 2

# References

<http://www.cs.rpi.edu/~szymansk/papers/han.10.pdf>

<https://en.wikipedia.org/wiki/Radial_basis_function_kernel>

<https://charlesmartin14.wordpress.com/2012/02/06/kernels_part_1/>

<https://www.cs.utah.edu/~jeffp/papers/gentleintroKD.pdf>