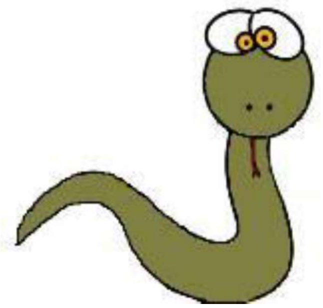
ACA PRESENTATION

Machine Learning

Ankit Bhardwaj

Anmol Porwal

1st Step: Learning Python



Major Sources : 1) Codeacademy
2)
https://openhatch.org/wiki/Intermediate_Python_Workshop/Projects
3)Python game(functional code available)

Introduction to awesome world of ML:

- https://www.coursera.org/learn/machine-learning
- https://class.coursera.org/nlp/lecture/preview

Topics covered:

- Types of learning: supervised and unsupervised.
- Important types of problems: Regression vs Classification Problems
- Linear Regression : Single variate and multi-variate
- Gradient Descent Algorithm
- Basic revision of linear algebra
- Logistic regression and Regularization
- Model fitting : underfitting and overfitting
- Precision and recall
- K-Mean Algorithm
- Dimensionality Reduction – PCA and SVD.

Natural Language Processing:

- http://textminingonline.com/dive-into-nltk-part-iv-stemming-and-lemmatization
- http://blog.nerdery.com/2013/03/playing-in-the-sandbox-building-a-spam-detector-with-python/
- Stemming and lemmatization
- Introduction to nltk

Moving towards the classifier

- Learning about the naïve bayes classifier:
- Algorithm, assumptions and related mathematics
- Discussions on better models
- Sources:
- https://class.coursera.org/machlearning001/lecture/241
- http://machinelearningmastery.com/naivebayesclassifierscratchpython/
- (functional code available)
- http://streamhacker.com/2010/05/10/textclassificationsentimentanalysis-naivebayesclassifier/
-

The Naïve Bayes Algorithm

- In the naïve bayes model of sentiment classification of movie reviews, the reviews are classified as pos and neg based on the words comprising the review as features. The core assumption is that all features are independent of each other and contribute independently to the probability of each class.
- $P(C|x).P(x)=P(x|C).P(C)$ [conditional probability equation where C is the corresponding class(pos or neg) and x is the feature at hand(word).] From joint probability model $P(C|x1,x2,\ldots.xN)$ is proportional to $P(C).P(x1|C)\ldots P(xN|C)$. Here $P(C)$ is half for both pos and neg while $P(x|C)$ is calculated on the basis of word frequency observed during training.
- During training, the classifier takes note of the words occurring in the movie reviews and stores them in a dictionary. It also counts the frequency of the word's occurring in both classes. Then P(word|pos) is calculated with feature probability and weighted probability.(taken almost directly from spam filter code)

- The probability for both the classes is calculated in the same way(using logarithm is better as the actual values are very low). These are not-normalized and hence do not add to 1. But since the problem at hand is of binary classification, direct comparison gives the prediction.
- The accuracy of the classifier varied around 80%.
- During the diabetes detection model from data we used gaussian model of naïve bayes classifier.

Face Recognition using Eigen Faces

- Sources:
- Research Paper Reading (Eigen Faces for Recognition – Mathew Turk and Alex Pentland, Vision and Modelling group, Media Library,MIT)
- Matlab tutorials online
- Dataset – Olivetti Research Laboratory(ORL) face database

Algorithm

- Principal Component Analysis
- Way of Dimentionality reduction
- Each image is converted into a vector of dimentions equal to number of pixels
- Now the size of this data can be reduced by changing the coordinate system
- This coordinate system is choosed in such a manner so that the variance of the data is maximum in these directions

- This is done by choosing these directions as the eigenvectors of the covariance matrix of the original images
- What is a covariance matrix ?
- If matrix A is the matrix with columns as the normalized coordinates of the original images.
- Then the matrix A * A transpose is the covariance matrix.
- Now dimention of the covariance matrix is very high so instead we calculate eigenvectors of a different matrix I. e. A transpose * A
- Not all eigenvectors are taken but a few with large eigenvalues

- The space created by these new eigenvectors is called face space.
- each image is Then represented by weights on these eigen faces.
- Then by faces of same individual a face class is created by averaging over weights.
- Now a test image belongs to a face class if the squared distance between its weights vector and face class weight vector is less than a predefined threshold
- We have to also check if the test image lies close to the face space .
- Thus we can create four different possibilities for any image .

Neural Networks

- The neural network model as the name suggests is based on human nervous system(network of neurons).
- The most basic computational unit is a neuron. It may be a percepteron, a sigmoid neuron or any other possible neuron. The names vary on the basis of the activaiton function of the neuron. Each neuron has a set of weights and a bias to calculate the output.
- Z=w.I+b[here, z is the input to the activation function, I is the input to the neuron, w is the vector of weights for corresponding inputs and b is the bias.]
- O=1/(1+e^-Z)[O is the output of the neuron. The function on the RHS is the sigmoid function. It can be replaced by any activation function but we generally tend to use the sigmoid fn as it is a smoothed out version of the step up fn]

- These neurons are then arranged into different layers and several interconnections are made. The entire thing comes under neural network architecture. The different architectures are used for different problems and the architecture design knowledge is said to be experience gained.
- The parameters (w and b) are determined by defining cost function and applying gradient descent. The algorithm used in the process is backpropagation.
- At 1st the weights and biases are initialized and then the output is calculated. Then the error in output(between predicted and actual) is propagated backward and the weights and biases of the network are updated.
- This process is continued till the error is minimised. This helps the neural network learn the weights and biases.
- This is done by the four fundamental equations of backpropagation. They are used for 1) finding error in output layer, 2) relating error in any layer to next layer error, 3) finding rate of change of cost wrt any bias and 4) weight.