Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 306 – 313

Procedia
Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

A Low Cost Implementation of Multi-label Classification Algorithm using Mathematica on Raspberry Pi

Neethu John[a,*], Surya R[a], Ashwini R[a], Sachin Kumar S[a], KP Soman[a]

[a] *Centre for Excellence in Computational Engineering and Networking*
Amrita Vishwa Vidyapeetham,
Coimbatore Campus, Tamilnadu-641112, India

Abstract

Implementation of data mining algorithms with low cost is one of the challenging tasks in the present world of massively increasing data. The key idea of this paper is to utilize the functionalities of Mathematica which is freely accessible on Raspberry Pi for the purpose of implementing Multi-label classification algorithm with low cost. With the facilities available in Mathematica software for Raspberry Pi, the line of code required for implementing data mining algorithms can be reduced sufficiently. Use of Random Kitchen Sink algorithm improves the accuracy of Multi-label classification and brings improvement in terms of memory usage for large dataset.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

Keywords: Multi-label classification; Mathematica; Raspberry Pi; Random Kitchen Sink; Data Mining.

1. Introduction

Mathematica, developed by Stephen Wolfram in the year 1988 has now become a powerful computational software that provide variety of applications in data mining, image processing and many other fields. Mathematica has proved to be very popular and prestigious tool for educational purposes all around the world. This computer language can

* Corresponding author. +91 85 4721 2821
E-mail address: jneethu.john@gmail.com

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.025

handle number theory problems, differential equations, graphing, visualization, 3-D modeling, image processing, stock market prediction, protein structure analysis, environment research etc. Ever since its release Mathematica has played a crucial role in many important discoveries and in the growth of financial modeling. It has become the foundation for large number of technical papers. By now Mathematica is being used by millions and adapted by many large universities around the globe. Recently Mathematica has been made free on Raspberry Pi which is a tiny version of computer. Raspberry Pi is highly affordable and plays a crucial role in the hobbyist market for computing. An extremely low power consumption, solid state storage, expansion capability, noise free behavior makes it a powerful tool for computational purpose. It marks the beginning of a new computing era.

Recently Multi-label classification has gained a great deal of importance in applications like demographic classification for classifying digital facial images, music categorization[1], and semantic scene classification[2]. In Multi-label classification, the specimens are related with more than one label $Y \subseteq L$. In the case of semantic scene classification, a photograph can be a member of more than one class like sunsets and beaches. Same song can be in more than one category for music categorization. In earlier days, Multi-label classification was primarily applicable in the field of medical diagnosis and text categorization. Text documents can be in multiple classes at the same time[1].

Efficiency of Multi-label classification can be enhanced by using Random Kitchen Sink (RKS) algorithm. Random Kitchen Sink is ideally suited for Radial Basis Function (RBF) kernels. RBF kernel is a Gaussian function and has several interesting properties. One important property of Gaussian function is that its Fourier Transform is another Gaussian. Secondly, this kernel is translation invariant. Random Kitchen Sink[3] speeds up the computation for a huge range of kernel functions. Random Kitchen Sinks are 100x faster and uses1000x less memory. These improvements, particularly in terms of memory usage, make the kernel methods more useful for applications that require real-time prediction and/or having large training sets.

Though, there exists several software implementations for Multi-label classification, this paper focuses on exploring the capabilities of Mathematica on Raspberry Pi for providing a low cost implementation of the same. Mathematical background of Multi-label classification and Random Kitchen Sink is explained in section 2. Hardware and software details are provided in section 3. Section 4 deals with the implementation steps in Mathematica. Results are explained in section 5. Section 6 concludes the paper.

2. Mathematical Background

2.1. Multi-label classification

Notations: Let n denotes the number of samples in the training data, k denotes the number of labels and d represents data dimensionality. $x_i \in R^d$ denotes the i^{th} observation and $y_i \in R^k$ encrypts its label information[4]. Let $X = [x_1, x_2, \ldots.., x_n] \in R^{d \times n}$ corresponds to the matrix of training data and $Y = [y_1, y_2, \ldots.., y_n] \in R^{k \times n}$ be the matrix of class label. We assume that both $\{x_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^n$ are centered, i.e., $\sum_{i=1}^n x_i=0$ and, $\sum_{i=1}^n y_i=0$. Identity matrix is denoted by I and e denotes the vector of ones. Forbenius norm of the matrix A is denoted by $\| A \|_F$ [4].

Here the aim is to find an unknown matrix W which maps our data matrix X to label matrix Y, where $W \in R^{k \times d}$. Lagrangian formulation of the problem can be expressed as:

$$L(W, \lambda) = \arg\min_w \| Y - WX \|_F^2 + \lambda \| W \|_F^2 \tag{1}$$

Where λ>0 is the regularization parameter which is included to avoid singularity of $X^T X$ and for preventing over fitting. The first term suggests that W is such that Y should be close to WX and the second term suggests that the elements of W should be small as far as possible.

$$L(W, \lambda) = \arg\min_w Tr[(Y - WX)^T (Y - WX)] + \lambda Tr(W^T W) \tag{2}$$

$$L(W, \lambda) = \arg\min_w Tr(Y^T Y - 2Y^T WX - X^T W^T Y + X^T W^T WX + \lambda W^T W) \tag{3}$$

Taking derivative and equating to 0, we get

$$\frac{\partial L}{\partial W} = -X^T Y - X^T Y + 2X^T XW + 2\lambda W = 0 \quad (4)$$

$$2(X^T X + \lambda I)W = 2X^T Y \quad (5)$$
$$W = (X^T X + \lambda I)^{\dagger} X^T Y \quad (6)$$

Where $(X^T X + \lambda I)^{\dagger}$ denotes the pseudo-inverse of $X^T X + \lambda I$. The index of maximum of projecting each of the test data on to the weight matrix W gives the label of that test data. One can ignore the bias term if both the targets and observations are centered. Then the projection matrix W is given as:

$$W = (X^T X)^{\dagger} X^T Y \quad (7)$$

2.2. Random Kitchen Sink algorithm

Random Kitchen Sink (RKS) algorithm is based on the fact that the Fourier Transform of Radial Basis Function (RBF) kernel can be interpreted as expectation of random variable. RBF kernel for data pair x and y is given by

$$k(x, y) = e^{-\sigma\|x-y\|^2} = e^{-\sigma(x-y)^T(x-y)} \quad (8)$$

Since, $k(x-a, y-a) = e^{-\sigma\|x-y\|^2}$ kernel is invariant of translation and hence

$$k(x, y) = k(x-a, y-a) = k(0, x-y) \quad (9)$$

So it can be represented as

$$k(x, y) = k(x-y) \quad (10)$$

It works as if it is a function of a single vector variable

$$d = x - y \quad (11)$$

One important property of Gaussian function is that its Fourier Transform is another Gaussian. Since Gaussian is symmetric, its transform is real. Here the function is a multivariate function given as $k(x, y) = k(d)$ and d is a vector. So one can think of the considered Fourier Transform is a multivariate Fourier Transform.

$$FT\left[k(x,y)\right] = FT\left[k(x-y)\right] \quad (12)$$

i.e; $FT\left(e^{-\sigma\|x-y\|^2}\right) = FT\left(e^{-\sigma\|d\|^2}\right) = FT\left(e^{-\sigma\left(d^T d\right)}\right) = \rho(\Omega)$ (13)

Here FT corresponds to the Fourier Transform and $\rho(\Omega)$ is multivariate normal. Ω has the same dimensionality as d. Then inverse of $\rho(\Omega)$ is given by

$$k(x, y) = k(x-y) = k(d) = \int_{\Omega} e^{j(x-y)^T\Omega} p(\Omega)d\Omega = \int_{\Omega} e^{j(d)^T\Omega} p(\Omega)d\Omega \quad (14)$$

The main observation to make here is that in RKS algorithm $\rho(\Omega)$ as a function is multivariate normal and therefore can be interpreted as a probability distribution. Also therefore the above expression can be interpreted as expected value of the multivariate function $e^{j(x-y)^T\Omega}$. The expected value means the average value. The above integration can therefore be thought to be performed by sampling several (say D) Ω vectors from multivariate normal distribution and then taking average of the function inside the integral over D samples. That is

$$\int_{\Omega} e^{j(d)^T\Omega} p(\Omega) d\Omega \approx \frac{1}{D}\sum_{i=1}^{D} e^{j\Omega_i^T(x-y)}\Omega_i - p(\Omega) \tag{15}$$

Here the problem of generating random vectors Ω_i from multivariate normal distribution arises. Assume the constituent variables in the vector are independent and are identically distributed. So if the dimension of the vector ω_i is n, then generate n random values from $N(0,1)$ and form the vector ω_i. Repeat this process D time and then evaluate the integral by averaging. The summation above can be split to obtain

$$\int_{\Omega} e^{j(d)^T\Omega} p(\Omega) d\Omega = \frac{1}{D}\sum_{i=1}^{D} e^{j\Omega_i^T x-y)} e^{-j\Omega_i^T y)}\Omega_i \tag{16}$$

$$k(x,y)=\frac{1}{D}\begin{pmatrix} e^{j(x-y)^T\Omega_1} \\ e^{j(x-y)^T\Omega_1} \\ . \\ e^{j(x-y)^T\Omega_D} \end{pmatrix} = \left\langle \begin{pmatrix} \frac{1}{\sqrt{D}} e^{jx^T\Omega_1} \\ \frac{1}{\sqrt{D}} e^{jx^T\Omega_2} \\ . \\ \frac{1}{\sqrt{D}} e^{jx^T\Omega_D} \end{pmatrix}, \begin{pmatrix} \frac{1}{\sqrt{D}} e^{-jy^T\Omega_1} \\ \frac{1}{\sqrt{D}} e^{-jy^T\Omega_2} \\ . \\ \frac{1}{\sqrt{D}} e^{-jy^T\Omega_D} \end{pmatrix} \right\rangle \tag{17}$$

i.e., $k(x-y)=E(e^{j(x-y)^T\Omega})$. Where Ω acts as a random variable. Since $k(x,y)=(\phi(x)^T)\phi(y)$, from the above expression one can infer that

$$\phi(x)=\begin{pmatrix} \frac{1}{\sqrt{D}} e^{jx^T\Omega_1} \\ \frac{1}{\sqrt{D}} e^{jx^T\Omega_2} \\ . \\ \frac{1}{\sqrt{D}} e^{jx^T\Omega_D} \end{pmatrix}, \phi(y)=\begin{pmatrix} \frac{1}{\sqrt{D}} e^{-jy^T\Omega_1} \\ \frac{1}{\sqrt{D}} e^{-jy^T\Omega_2} \\ . \\ \frac{1}{\sqrt{D}} e^{-jy^T\Omega_D} \end{pmatrix} \tag{18}$$

This means that we can create finite (D) dimensional $\phi(x)$ by the following steps.

- Create D number of n-dimensional random Gaussian distributed vectors and $\Omega_i s$.
- Compute $\phi(x)$ as per the above expression.

But $\phi(x)$ computed above is a vector with complex elements. We can avoid dealing complex numbers by creating another vector with two times D elements. First D elements are cosines and next D elements are sine values of $\Omega^T x$. Thisdoes not affect the value of $k(x, y)$. Once we have $\phi(x)$ for all data points, we can use any linear classifier for classification jobs.

3. Hardware and Software

The hardware which is used here is Raspberry Pi or Pi in short. It is a fully blown low cost desktop PC which can be directly connected to the internet and is able to display videos with high definition. Since Pi runs on Linux, there is no need to pay for an OS. By using Raspberry Pi we are able to choose from a wide range of programming languages for implementing our project. Raspberry Pi is a flexible platform for fun utility and experimentation. Mathematica provides an integrated platform for technical computing. It consists of essentially two programs, which communicate with each other and combined to form an interactive interpreter: the front-end provides the graphical user-interface and the kernel for the calculations. The front-end takes the role of a progressive text processor. Creation of notebooks, portable documents are carried out by the front-end. Mathematica which is the software used here has become free on Raspberry Pi. This made our project a low cost implementation in data mining algorithms.

4. Implementation

Here we perform Multi-label classification on iris dataset. The iris data consist of three classes, each containing 50 samples. Each sample is described by four attributes: the length and width of sepals and petals. The fifth attribute denotes the class (species) of the iris flower. The objective of this procedure is to find which species an unclassified flower will belong to. From each class 45 out of 50 samples are taken as train data, remaining five samples are retained for testing. Following are the implementation steps in Mathematica.

- Load the training dataset to variable X and arrange it in such a way that each column represents an instance in the dataset.
  Example:
  X=Import["iristrain.xlsx"]
  $X=X^T$
- Take the transpose of X and find X^TX
  $Q=X^T.X$
- Value of regularization parameter λ is obtained by
  lmd=Max[Q]+5
- Import the label matrix to variable Y.
  Example:
  Y=Import["label.xlsx"]
- Weight matrix W is obtained as:
  $\mathrm{W = PseudoInverse[(IdentityMatrix[15] * lmd) + }Q\mathrm{].X^T.Y}$
- Import the test data to variable A.
  Example:
  A=Import["iristest.xlsx"]
- Project test data A on to the weight matrix W.
  $P=W^T.A$
- Label of test data is obtained by the following steps:
  g=Max/@Transpose[P]
  h={}
  For[u=1,u≤number of instances in test dataset, u=u+1,Print[h =AppendTo[h,First[Position[P,Part[g,u]]]]]]

Efficiency of this method can be obtained by comparing the actual label of test data and the label obtained from 8th step. Here the procedure of Multi-label classification with ridge regression is followed. By excluding 3rd step and neglecting the lambda term in 5th step one can obtain Multi-label classification without ridge regression. i.e., step 5 becomes W=PseudoInverse[X].X^T.Y

For improving efficiency of classification one can go for Random Kitchen Sink method. For applying RKS before going to 2nd step the following steps need to be performed.

- The following command create pseudo random valued matrix. These values are obtained from the standard normal distribution.
  phi = RandomReal[{-3,3},{80,First[Dimensions[X]]}]
- This step creates a new dataset as a result of applying RKS
  k=Last[Dimensions[X]]
  For[q=1,q≤k,q=q+1,X=Transpose[Insert[Transpose[X],phi.N[Cos[X[[All,q]]]],Last[Dimensions[X]]+1]]]
  For[q=1,q≤k,q=q+1,X=Transpose[Insert[Transpose[X],phi.N[Sin[X[[All,q]]]],Last[Dimensions[X]]+1]]]
  X=X[[All,k+1;;Last[Dimensions[X]]]]

Another observation is that instead of going for RKS, simply attaching sine and cosine value of each column of dataset as a new column to the existing data and performing classification will also give better result. Performing the subsequent steps after step 1 of Multi-label classification will realize the method mentioned above.

k=Last[Dimensions[X]]

For[q=1,q≤k,q=q+1,X=Transpose[Insert[Transpose[X],N[Sin[X[[All,q]]]],Last[Dimensions[X]]+1]]]

For[q=1,q≤k,q=q+1,X=Transpose[Insert[Transpose[X],N[Cos[X[[All,q]]]],Last[Dimensions[X]]+1]]]

5. Result and Discussion

Multi-label classification without and with ridge regression are implemented here. For improving efficiency, Random Kitchen Sink procedure is applied. Classification is performed on the iris, Tamil OCR and medical multi-label data sets[5]. The information about the dataset such as the number of specimens and the number of class labels are given in Table1.

Table 1. Examples, class labels and attributes of datasets.

Dataset	Train	Test	Number of classes	Number of Attributes
Iris	135	15	3	4
Tamil OCR	300	54	6	7
Medical	75	13	2	7

Confusion matrices obtained by performing Multi-label classification on the three datasets are listed below. Result of performing Multi-label classification with and without applying RKS are organized as separate confusion matrices.

Table 2. Tamil OCR data classification without RKS.

		Predicted					
		Label 1	Label 2	Label 3	Label 4	Label 5	Label 6
Actual	Label 1	9	0	0	0	0	0
	Label 2	0	7	2	0	0	0
	Label 3	0	1	8	0	0	0
	Label 4	0	0	0	9	0	0
	Label 5	0	0	0	0	9	0
	Label 6	0	0	0	0	0	9

Table 3. Iris data classification without RKS.

		Predicted		
		Label 1	Label 2	Label 3
Actual	Label 1	5	0	0
	Label 2	0	4	1
	Label 3	0	0	5

Table 4. Medical data classification without RKS.

		Predicted	
		Label 1	Label 2
Actual	Label 1	8	0
	Label 2	5	0

Table 5. Iris data classification with RKS.

		Predicted		
		Label 1	Label 2	Label 3
Actual	Label 1	5	0	0
	Label 2	0	5	0
	Label 3	0	0	5

Table 6. Medical data classification with RKS.

		Predicted	
		Label 1	Label 2
Actual	Label 1	8	0
	Label 2	0	5

Table7. Tamil OCR data classification with RKS.

		Predicted					
		Label 1	Label 2	Label 3	Label 4	Label 5	Label 6
Actual	Label 1	9	0	0	0	0	0
	Label 2	0	9	0	0	0	0
	Label 3	0	0	9	0	0	0
	Label 4	0	0	0	9	0	0
	Label 5	0	0	0	0	9	0
	Label 6	0	0	0	0	0	9

6. Conclusion and Future Work

Low cost implementation of Multi-label classification is attained by bringing together the capabilities of Mathematica and Raspberry Pi. Using Mathematica, which has become freely accessible on Raspberry Pi, classification can be performed within few lines of code. Inclusion of Random Kitchen Sink procedure brings improvements, particularly in terms of memory usage for applications that requires real-time prediction and/or having large training sets.

In this project only the Multi-label classification algorithm of data mining is implemented using Mathematica on Raspberry Pi. It is possible to make use of the capabilities of Mathematica and Raspberry Pi for performing the low cost implementation of various data mining algorithms. The machine learning framework for Mathematica, as its name indicates, a collection of algorithms for machine learning can be used to perform data analysis. It provides the users the ability to interactively analyze their data, with short cycles of examining the results and changing parameter settings.

References

1. Li, Tao, Mitsunori Ogihara. Detecting emotion in music. *ISMIR*; 2003.
2. Boutell, Matthew R, et.al. Learning multi-label scene classification. *Pattern recognition* 37.9 : 1757-1771; 2004.
3. Rahimi, Ali, Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems;* 2007.
4. Sun, Liang, Shuiwang Ji, Jieping Ye. A least squares formulation for canonical correlation analysis. *Proceedings of the 25th international conference on Machine learning.* ACM; 2008.
5. KP Soman, R Loganathan, V Ajay. Machine Learning with SVM and Other Kernal Methods. PHI Learning Private Limited, New Delhi; 2009.
6. Grigorios Tsoumakas, Ioannis Katakis. Multi label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3); 2007.
7. Sugiyama, Masashi, et.al. Least-squares probabilistic classifier: A computationally efficient alternative to kernel logistic regression. *Proceedings of International Workshop on Statistical Machine Learning for Speech Processing (*IWSML*)*, Kyoto, Japan; 2012.
8. S Wolfram. Mathematica: A System for doing Mathematics by Computer, *Addison Wesley Publishing Company*, Inc, 2nd edition.
9. Bi, Wei, James Kwok. Efficient Multi-label Classification with Many Labels. *Proceedings of the 30th International Conference on Machine Learning* (ICML); 2013.
10. Raspberry Pi official site. http://www.raspberrypi.org – Accessed on April 2; 2014.
11. Tsoumakas, Grigorios, Ioannis Katakis, Ioannis Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. *Proc. ECML/PKDD Workshop on Mining Multidimensional Data (MMD)*; 2008.