Implementation of Multi-Label Classification in Sparse Matrices

Napoleon Maraidonis[1] under the supervision of Michalis Titsias[1]

1 Department of Informatics, Athens University of Economics and Business

Abstract

Introduction

Introduction [1] [2] [4]

Notation

- Lowercase bold letter denotes a n * k matrix where n is rows and k is columns : $\mathbf{w} \in \mathbb{R}^{n*k}$
- Lowercase bold letter with subscript denotes the n-nth row of the matrix : $\mathbf{x}_n \in \mathbb{R}^k$
- Lowercase letter will denote a real number unless stated otherwise : $x \in \mathbb{R}$
- Uppercase Italics letter denotes a set : T
- The letter $\hat{y}$ will denote predicted label(s).
- The letter X will denote the train data matrix which will have its first column, full of ones in addition to other data, $X \in \mathbb{R}^{N*K}$ with N the number of instances and K the number of features.
- The letter Y will denote train data true labels.
- $sigm(x) \triangleq \frac{1}{1+exp(x)}$

Classifier Chains

Classifier Chains is a transformation of the problem which takes into account label dependence. Classifier Chains works in the following way [4]:

$\hat{y}_1 = h_1(X),\ X = \left[X|\hat{y}_1\right],$
$\hat{y}_2 = h_2(X),\ X = \left[X|\hat{y}_2\right],$
...
$\hat{y}_n = h_n(X)$

Each classifier is trained using results from previous classifiers in a chain, the order which labels are predicted is arbitary. There are more complicated schemes that aim to optimise the order that labels are predicted such as Bayes Optimal CC [4].

Binary Logistic Regression Classifier

Logistic Regression Classifier is a classification algorithm described by these expressions (implying two classes):

$\hat{y} = \underset{y\in\{0,1\}}{argmax}\ p(y|\mathbf{x})$ where $p(y=1|\mathbf{x}) = sigm(\mathbf{w}^T\mathbf{x})$

The joint distribution of both classes is $p(Y|X,\mathbf{w}) = \prod_{n=1}^{N}\ p(Y_n|\mathbf{x}_n,\mathbf{w})$ so the logarithmic likelihood is
$L(\mathbf{w}) = \sum_{n}^{N}\ Y_n\ log(sigm(\mathbf{w}^T\mathbf{x}_n)) + (1-Y_n)\ log(1-sigm(\mathbf{w}^T\mathbf{x}_n))$. In order to find optimal $\mathbf{w}$ its cost function $-L(\mathbf{w})$ will be optimized utilising stochastic gradient descent.

Optimization of the cost function

Gradient descend is an iterative optimization algorithm which in each iteration the weights are updated according to the formula: $\mathbf{w}^{(k+1)} \leftarrow \mathbf{w}^{k} + l\ \nabla L(\mathbf{w}^{(k)})$ where l is the learning rate.

The computational complexity of calculating ∇L is increasing according to the number of training instances. In order to generalize well in any machine learning algorithm, a great number of training examples is needed and gradient descend becomes computationally prohibitive.

Stochastic gradient descend is a variation of gradient descend as its name suggests, uses sampling in order to circumvent the computational cost.

$\nabla_w L(\mathbf{w}) = -\mathbf{x}^T\mathbf{t} + \mathbf{x}^T\mathbf{x}\mathbf{w}$

Supposing a batch B choosen randomly from the training data with a size of b

The update procedure becomes like this :

$B \leftarrow choose\ b\ training\ instances\ from\ each\ update$
$\mathbf{w}^{(k+1)} \leftarrow \mathbf{w}^{k} + l\ \frac{1}{B}\sum_{b=0}^{B-1}\nabla L(\mathbf{w}^{(k)},\mathbf{x}(b))$

Prototype Structure and Documentation

General Structure

The prototype consists of three main parts : (a) Binary Classification Algorithm, (b) Multi-Label Classifiactation Interface (c) Score function. More specifically :

a Firstly, the implementation of any Binary Classification Algorithm will be required: multiclass logistic regression was chosen for simplicity. The sparsity of the training data will be put into account while implementing the algorithm and stochastic gradient descent will be used for the optimization of the loss function.

b Classifier chains as explained above will be used.

c Scoring will be made according to the formula:

$$accuracy \triangleq \frac{|T \cap P|}{|T \cup P|}$$

Notes

- The implementation of any Multiclass Classification Algorithm will follow the following contract : The methods train(Xtrain,Ytrain), and predict(Xtest) will be implemented.
- The implementation of any Multy Label Classifiactation Interface will follow the following contract using the implemented Multiclass Classification Algorithm : The methods train(Xtrain,Ytrain), and predict(Xtest) will be implemented.

Results

Discussion

Acknowledgments

References

1. *Multiclass-Multilabel Classification with More Classes than Examples*. AISTATS, 2010.

2. Y. P. M. V. Kush Bhatia, Himanshu Jain. The extreme classification repository: Multi-label datasets and code.

3. J. Read. Multi-label classification. Technical report, Universidad Carlos III de Madrid. Department of Signal Theory and Communications, 2013.

4. J. Read. Multi-label classification. Technical report, Department of Information and Computer Science Helsinki, Finland, 2015.