Kernel Logistic Regression Algorithm for Large-Scale Data Classification

Murtada Elbashir[1] and Jianxin Wang[2]
[1]Faculty of Mathematical and Computer Sciences, University of Gezira, Sudan
[2]School of Information Science and Engineering, Central South University, China

Abstract: *Kernel Logistic Regression (KLR) is a powerful classification technique that has been applied successfully in many classification problems. However, it is often not found in large-scale data classification problems and this is mainly because it is computationally expensive. In this paper, we present a new KLR algorithm based on Truncated Regularized Iteratively Re-weighted Least Squares(TR-IRLS) algorithm to obtain sparse large-scale data classification in short evolution time. This new algorithm is called Nystrom Truncated Kernel Logistic Regression (NTR-KLR). The performance achieved using NTR-KLR algorithm is comparable to that of Support Vector Machines (SVMs) methods. The advantage is NTR-KLR can yield probabilistic outputs and its extension to the multi class case is well defined. In addition, its computational complexity is lower than that of SVMs methods and it is easy to implement.*

Keywords: *KLR, IRLS, nystrom method, newton's method.*

Received November 18, 2012; accepted July 2, 2013; published online September 4, 2014

1. Introduction

In many data mining applications in a wide range of fields, including social science, bioinformatics, etc., there is a need for large-scale data classification algorithm in order to derive useful information out of these data. There are a variety of supervised learning techniques devised for the purpose of classification. Among these techniques are the kernel based methods such as Support Vector Machines (SVMs) and Kernel Logistic Regression (KLR). Based on recent advances in statistical learning theory, SVMs deliver state of the art performance in real world applications [3, 7]; its training algorithm builds a model that assigns new examples into one category or the other to reach classification of the input variable. The SVMs model is a representation of the examples as points in space, mapped so that, the examples of the separate categories are divided by a clear gap that is as wide as possible. The SVM method requires solving a constrained quadratic optimization problem with a time complexity of $O(n^3)$ where n is the number of training instances. Iterative chunking method, which divides the overall problem into small active training set, was designed to implement SVMs in large scale data sets. The extreme form of chunking is the Sequential Minimal Optimization (SMO) [19]. LIBSVM, which is the state of the art toolbox [2], uses SMO solver described in [4]. The problem of LIBSVM is that sometimes the training time for large-scale data sets is unrealistic.

On the other hand, KLR has also proven to be a powerful classifier [1, 12, 23]. KLR is a kernel version of logistic regression, which is a well-known classification method in the field of statistical learning. In KLR, the input vector is mapped to a high-dimensional space (feature space). Unlike SVMs, KLR includes the probabilities of occurrences as a natural extension. More over KLR can be extended to handle multi class classification problems and it requires solving only unconstrained optimization problem [8, 14]. KLR is fitted using maximum likelihood, it also has time complexity of $O(n^3)$ and is often not found on predicting large-scale data due to its computational demand [14].

Iteratively Re-weighted least Squares (IRLS) that implement newton-raphson method is applied effectively for finding the maximum likelihood estimates of a LR model [6]. Komarek and Moore [16] modified IRLS that mimics truncated Newton's methods and added regularization. They named their algorithm Truncated Regularized IRLS (TR-IRLS), and they show that it can be effectively implemented to classify large scale data using LR and that it can outperform the SVMs algorithm. Later on, another type of truncated newton and truncated newton interior point methods [15] called trust region newton's method [17] is applied for large-scale LR problems. However, LR linearity may be an obstacle to handling highly nonlinearly separable data sets [16].

Maalouf *et al*. [18] combine the speed of the TR-IRLS algorithm with the accuracy generated by the use of kernels for solving nonlinear problems. Their kernel version of the TR-IRLS algorithm, which they named Truncated Regularized KLR (TR-KLR) is just as easy to implement and requires solving only an unconstrained regularized optimization problem. It can be applied successfully for small to medium size data classification problems. However, for large scale problems TR-KLR still becomes computationally expensive.

In this paper, we present a new practical and scalable algorithm based on TR-KLR algorithm to be used for large scale data classification. This new algorithm is called Nystrom Truncated KLR (NTR-KLR). In NTR-KLR, we adopt eigen-decomposition of the kernel matrix into eigenvalues/eigenvectors matrices and then select the first $p<<n$ eigenvalues/eigenvectors form these matrices to approximate the kernel matrix. We used nystrom method to reduce the computation cost of computing the eigen-decomposition by selecting small sample of size $m<<n$ from the feature's matrix using k-means clustering with outlier removal. The performance achieved using the proposed NTR-KLR is comparable to that of SVMs methods. The advantage is that NTR-KLR can yield probabilistic outputs and it can be extended to handle multi class classification. In addition its computational complexity is lower than that of SVMs methods and it is easy to implement.

2. Material and Methods

2.1. Kernel Logistic Regression

KLR is the kernel version of logistic regression, which is a well-known statistical model for classification. Unlike LR, KLR enables the classification of linearly non-separable problems by transferring the input features to a higher dimensional space, via the kernel trick. The kernel is a transformation function that must satisfy mercer's necessary and sufficient conditions, which state that a kernel function must be expressed as an inner product and must be positive semi-definite. The logit link function of the KLR can be written as follows [18]:

$$\eta_i = \varphi(x_i)\beta \tag{1}$$

Where β is the LR parameter, The transformation $\phi(.)$ is often unknown but the dot product in the feature space can be expressed in terms of the input vectors through the kernel function so, the log it link function can be given as follows [18]:

$$\eta_i = k_i \alpha \tag{2}$$

Where k_i is the i^{th} row in the kernel matrix and α is the KLR parameters. Each row k_i is modelled with its corresponding class as follows:

$$p_i = \frac{e^{k_i \alpha}}{1 + e^{k_i \alpha}} \tag{3}$$

The penalized log-likelihood can be rewritten with respect to α as:

$$ln\, L(\alpha) = \sum_{i=1}^{n} (y_i \ln p_i + (1 - y_i) ln(1 - p_i)) - \frac{\delta}{2} \alpha^T k \alpha \tag{4}$$

Maximizing the log-likelihood above is equivalent to minimizing the deviance [9], which is the negative of the log-likelihood and is given as follows:

$$DEV(\alpha) = -2\, ln(\alpha) \tag{5}$$

The KLR can be fitted by the solution of a convex optimization problem, which can be solved efficiently via the algorithm IRLS. In IRLS, each iteration finds the Weighted Least Squares (WLS) estimates for a given set of weights, which are used to construct a new set of weights [16]. The WLS problem for the KLR in each iteration is given as follows:

$$(K^T W K + \delta K)\hat{\alpha}^{(c+1)} = K^T W z^{(c)} \tag{6}$$

Where W is the weight matrix, K is the kernel matrix, and z is the vector of adjusted response, which is given as follows:

$$z^{(c)} = K\hat{\alpha}^{(c)} + V^{-1}(y - p) \tag{7}$$

Maher *et al*. [18] applied an algorithm based on TR-IRLS, which they called TR-KLR. In TR-KLR the linear system in Equation 6 for each iteration can be solved using the Conjugate Gradient (CG) method after specifying an initial estimate $\hat{\alpha}^{(0)}$. To avoid the long computations that CG may suffer from, a limit to the number of CG iterations can be placed, thus creating an approximate or truncated newton direction [18]. The TR-KLR is suitable for small to medium size data sets, but for large-scale data e.g., n>10,000 it is computationally expensive, because the linear system in Equation 6 must be solved for each Newton's iteration. Also, storing the kernel matrix for such large data is prohibitive on modern workstations (although, this boundary can be pushed further by using high-performance computers).

2.2. The NTR-KLR Algorithm

To extend the usability of TR-KLR for large data sets, we can adopt eigen-decomposition of the kernel matrix K in the form.

$$K_{n \times n} = U_n \Lambda_n U_n^T \tag{8}$$

Where $\Lambda_n = diag(\lambda_i)$ and $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$ are the eigenvalues of the matrix K, U_n is the matrix of the eigenvectors that correspond to the eigenvalues and n is the number of the data points. We can select the first p eigenvectors and eigenvalues form the matrices U and Λ respectively, where $p<<n$, to approximate the kernel matrix. This approximation is motivated by it is widely usage e.g., principal component analysis [21]. Using this approximation can reduce the computational cost drastically. However, computing the eigen decomposition itself is also computationally expensive. To reduce the computational cost of computing the eigen decomposition we can select small sample of size $m<<n$ from the features matrix [20] to construct the following eigen problem:

$$K_{m \times m} = U_m \Lambda_m U_m^T \tag{9}$$

Now, we can extend the eigenvalues/eigenvectors of the $K_{m \times m}$ to the all points using the following nystrom approximation [21]:

$$\tilde{\lambda}_i^{(n)} = \frac{n}{m}\lambda_i^{(m)}, \quad u_i^{(n)} = \sqrt{\frac{m}{n}}\frac{i}{\lambda_i^{(m)}} K_{n,m} u_i^{(m)} \tag{10}$$

Where $\lambda_i^{(m)}$ and $u_i^{(m)}$ are the i^{th} eigenvalue/ eigenvector of the $m \times m$ eigenproblem and $K_{n \times m}$ is the appropriate $n \times m$ sub matrix of K. The quality of approximation for the kernel matrix K can be considered at the m points used for eigen decomposition and the other n-m points. In general we have a choice of how many of the approximate eigenvalues/vectors to include in our approximation of K; choosing the first p we get:

$$\tilde{K} = \sum_{i=1}^{p} \tilde{\lambda}_i^{(n)} \tilde{u}_i^{(n)} (\tilde{u}_i^{(n)})^T \tag{11}$$

By using nystrom method and setting $p=m$ it is easy to show that the approximation $\tilde{K}$ for the kernel matrix is as follows [21].

$$\tilde{K} = K_{n \times m} K_{m \times m}^{-1} K_{m \times n} \tag{12}$$

By using the approximation above, we don't need to compute and store the whole kernel matrix in the memory but only a m by n portion of it which can help in utilizing the memory usage. The selected $m << n$ from the features matrix should minimize the mean squares error or in another words it should contain as much information as possible. Because the nystrom low-rank approximation depends crucially on the quantization error induced by encoding the sample set with landmark points, one can simply use the clusters obtained with (k-means algorithm) as a selected vectors [13, 24]. In another words the k-means clustering algorithm can be used to select m vectors from the features matrix using the clustering approach. k-means clustering algorithm is susceptible to outliers, these outliers can affect the quality of information contained in the m selected vectors and in this case k-means clustering algorithm with outlier's removal can contain as much information as possible and give high classification accuracy. The computational time of the KLR using nystrom approach scales to $O(nm^2)$ [21].

The following algorithm represents the NTR-KLR algorithm, which extend the usability of TR-KLR to large data sets. The algorithm terminates when the relative difference of deviance between two consecutive iterations is not greater than a specified threshold.

Algorithm 1: NTR-KLR MLE using IRLS

Input: X, y
Result: $\hat{\alpha}$

begin
Select m vectors from X using k-means clustering to construct the eigen problem in Equation 9.
Extend the eigenvectors/values to all points as in Equations 9.
Approximate the kernel matrix $\tilde{K}$ *as in Equation 11.*
Determine initial estimate for $\hat{\alpha}^{(0)}$.
$c=0$
While $\left|\frac{DEV^{(c)} - DEV^{(c+1)}}{DEV^{(c+1)}}\right| > \varepsilon$ *and* $c <$ *IRLSmax iteration*
for i=1 to m do

$$\hat{p}_i = \frac{1}{1 + e^{-\tilde{k}_i \hat{\alpha}}}$$
$$v_i = \hat{p}_i (1 - \hat{p}_i)$$
$$z_i = \tilde{k}_i \hat{\alpha}^{(c)} + \frac{(y_i - \hat{p}_i)}{\hat{p}_i (1 - \hat{p}_i)}$$

End for
$V = diag(v_1, \ldots, v_n)$
$(\tilde{K} V \tilde{K} + \delta \tilde{K}) \hat{\alpha}^{(c+1)} = \tilde{K}^T V z^{(c)}$ */* solve using Algorithm 2*
$c=c+1$
End while
End

Algorithm 2: Linear CG for solving the linear system AX=b

Input: $A = (\tilde{K} V \tilde{K} + \delta \tilde{K})$, $b = \tilde{K}^T V z^{(c)}$, $\hat{\alpha}^{(0)}$
Result: $\hat{\alpha}$

begin
$r^{(0)} = b - A\hat{\alpha}^{(0)}$
$c=0$
while $\left\| r^{(c+1)} \right\|^2 > \varepsilon$ *and* $c \leq$ *CGmax iteration*
If c=0 then
$\xi^{(c)} = 0$
Else
$\xi^{(c)} = \frac{r^{T(c+1)} r^{(c+1)}}{r^{T(c+1)} r^{(c)}}$
End if
$d^{(c+1)} = r^{(c+1)} + \xi^{(c)} d^{(c)}$
$s^{(c)} = \frac{r^{T(c)} r^{(c)}}{d^{T(c)} A d^{(c)}}$
$\hat{\alpha}^{(c+1)} = \hat{\alpha}^{(c)} + \xi^{(c)} d^{(c+1)}$
$r^{(c+1)} = r^{(c)} - s^{(c)} A d^{(c+1)}$
$c=c+1$
End while
End

2.3. Data Sets

The benchmark datasets that are used to measure the performance of NTR-KLR were eight datasets. All these datasets contain more the 10,000 instance. We consider only binary classification, because it facilitates the use of SVM and allows us to compare it fairly with NTR-KLR. Adults, ijcnn1 and w6a were originally binary class data sets, but shuttle, connect-4, nursery, seismic and protein were multi-class data sets; we convert them to binary class, so as to classify one class against the rest of the classes. These sets are downloaded from the LIBSVM datasets, and the UCI machine learning repository. The number of features, and instances, for each dataset is shown in Table 1.

Table 1. The large-scale data sets used to measure the performance of NTR-KLR.

Data Set	Instances	Features
Adults (a9a)	32561	123
ijcnn1	49990	22
W6a	17188	300
Shuttle	43500	9
Connect-4	67557	42
Nursery	12960	8
Seismic	78823	51
Protein	17766	357

To evaluate the performance of NTR-KLR algorithm on small size data sets and to compare it with TR-KLR and SVM algorithms we used the data sets that are shown in Table 2. These datasets are also downloaded

from the LIBSVM data and the UCI machine learning repository.

Table 2. The small-size data sets used to measure the performance of NTR-KLR.

Data Set	Instances	Features
Heart	270	13
Liver	345	6
Survival	306	3
Australian	690	14
Diabetes	768	8
Ionosphere	351	34

2.4. Performance Measures

The quality of prediction is evaluated using the following measures: Accuracy, Matthews Correlation Coefficient (MCC), precision, sensitivity, specificity, and the ROC curve. These measures are consistent measures that are used to evaluate machine learning methods. Let True Positives (TP) be the number of correctly classified instances as 1, True Negatives (TN) be the number of correctly classified instances as 0, False Positives (FP) be the number of instances incorrectly classified as 1 and False Negatives (FN) be the number of instances incorrectly classified as 0. The prediction accuracy, which is defined as the percentage of correctly classified residues is calculated as follows:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{13}$$

The MCC can be calculated as:

$$MCC = \frac{TP*TN - FP*FN}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}} \tag{14}$$

The probability of correct prediction or precision, it is also called Predicted Positive Value (PPV) and it is given as follows:

$$Precision = \frac{TP}{TP+FP} \tag{15}$$

Sensitivity or coverage is the percentage of correctly predicted instances among the observed instances or it is the fraction of the total positive samples that are correctly predicted and it is given as follows:

$$Sensitivity = \frac{TP}{TP+FN} \tag{16}$$

Specificity is the fraction of total negative samples that are correctly predicted.

$$Specificity = \frac{TN}{TN+FP} \tag{17}$$

The classification results from the most positive classification to the most negative classification are plotted using the ROC curve plots, it is considered to be a comprehensive evaluation of classifier performance. From the ROC curve we can obtain the area under curve AUC. A high AUC value indicates good classification performance.

We have added another three measures, which have been used in medical diagnosis to analyse tests [11], these measures are Youden's index, Likelihoods measure and the diagnostic odds ratio. These measures will bring in new characteristics, which include the confirmation capability with respect to classes, that is, the estimation of the probability of the correct predictions of positive and negative labels; and the ability to avoid failure, namely, the estimation of the complement of the probability of failure. Furthermore, these measures are easily comparable.

Youden's index γ [22] evaluates the algorithm's ability to avoid failure or the ability of an algorithm to correctly labels examples. It equally weights the algorithm's performance on positive and negative examples:

$$\gamma = sensitivity + specificity - 1 \tag{18}$$

A higher value of γ indicates better ability to avoid failure.

Likelihoods measure, it is a measure that combines both sensitivity and specificity and can evaluate classifier's performance to a finer degree with respect to both classes. It is calculated with respect to the following two measures.

$$\rho+ = \frac{sensitivity}{1- specificity}, \rho- = \frac{1- sensitivity}{specificity} \tag{19}$$

A higher positive likelihood and a lower negative likelihood mean better performance on positive and negative classes respectively. The relation between the likelihood of two algorithms A and B establishes which algorithm is preferable and in which situation. This relationship is given as follows:

$\rho+^A > \rho+^B$ and $\rho-^A < \rho-^B$ implies that A is superior overall; $\rho+^A < \rho+^B$ and $\rho-^A < \rho-^B$ implies A is superior for confirmation of negative examples; $\rho+^A > \rho+^B$ and $\rho-^A > \rho-^B$ implies A is superior for confirmation of positive examples; $\rho+^A < \rho+^B$ and $\rho-^A > \rho-^B$ implies A is inferior overall.

The Diagnostic Odds Ratio (DOR) [5] is also a global performance measure. It has been used as a measure of diagnostic discrimination in medicine to test the ratio of the odds of positivity in disease relative to the odds of positivity in the non-diseased. DOR can also be used in machine learning to compare algorithms performances. It is calculated as follows:

$$DOR = \frac{sensitivity/(1- sensitivity)}{(1- specificity)/specificity} \tag{20}$$

The value of a DOR ranges from 0 to infinity, with higher values indicating better discriminatory test performance. A value of 1 means that the test does not discriminate between two algorithms. Values lower than 1 point to improper test interpretation. The Wilcoxon signed ranks test, which is a non-parametric statistics is used for analysing the differences between NTR-KLR and LIBSVM algorithm over the

benchmark datasets. Wilcoxon signed ranks test ranks the differences in performances of two classifiers for each data set, ignoring the signs and compares the ranks for the positive and the negative differences. The differences are ranked according to their absolute values; average ranks are assigned in case of ties. Let R^+ be the sum of ranks for the data sets on which the second algorithm outperformed the first and R^- the sum of ranks for the opposite. Ranks of di=0 are split evenly among the sums; if there is an odd number of them, one is ignored:

$$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i)$$
$$R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \quad (21)$$

Let T be the smaller of the sums, T= $min(R^+, R^-)$. Most the statistical books include a table of exact critical values for T from 1 up to 25 degree of freedom (or sometimes more). For a larger number of data sets, the statistics:

$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}} \quad (22)$$

Could be used, which is distributed approximately normally. With a=0.05, the null-hypothesis can be rejected if z is smaller than −1.96.

3. Experimental Setup

We compared the performance and computational time of NTR-KLR with the SVM method. To determine the optimal parameters for NTR-KLR, ten folds cross validation was performed on all the datasets. That is, these sets were randomly divided into ten subsets, each containing approximately equal number of instances. Nine of the ten subsets were merged together to form a training set that will be used to train the NTR-KLR method. The unused subset is used for testing. This process was repeated ten times to test the prediction result for each testing set. The best values were determined for each training set. The training set with its associated best values was used to construct the NTR-KLR model. For the SVM we used the grid search method, which is explained in [10] to tune its parameters. The following Radial Basis Function (RBF) is used as a kernel function for both the SVM and NTR-KLR.

$$K(x_i, x_j) = e^{(-\frac{1}{2\sigma^2}\|x_i - x_j\|)^2} \quad (23)$$

All of the datasets were scaled to values between 1 and 0. All of the computations for NTR-KLR and SVM were carried out using MATLAB version 2009 a on a 3 GB RAM computer. For the SVM method, we used LIBSVM toolbox [2] for MATLAB. For NTR-KLR the maximum number of newton raphson's iterations is set to 50, while the maximum number of iteration for the CG is set to 200.

4. Results

It can be observed from Tables 3 and 4 that the MCC of NTE-KLR is noticeably better than that of SVM on ijcnn1 and shuttle datasets and it is slightly better than SVM on adults, nursery and seismic data sets. The MCC of SVM is better than NTR-KLR connect, w6a, and protein data sets. The accuracy of NTR-KLR is better than that of SVM on adult, ijcnn1, shuttle and nursery data sets and it is worse on connect, seismic, and protein data sets. The differences in the accuracy on these data sets are not significant. The precision of NTR-KLR is noticeably better than SVM on ijcnn1 and it is worse on protein data sets. The precision on the other data sets is almost the same. The sensitivity of the NTR-KLR is better than that of SVM on four out of eight data sets; it is the same in one data set and less that SVM in three data sets.

Table 3. The performance results of the NTR-KLR.

Data Set	MCC	Accuracy	Precision	Sensitivity	Specificity	AUC	Uind	*P* +	*P* -	DOR
Adults	0.56	0.86	0.71	0.59	0.93	0.90	0.52	8.09	0.44	18.39
Connect	0.41	0.77	0.82	0.87	0.52	0.81	0.39	1.81	0.25	7.28
Ijcnn1	0.67	0.95	0.79	0.61	0.98	0.99	0.59	38.17	0.40	95.22
Shuttle	0.99	0.99	0.99	0.99	0.99	0.99	0.99	159.74	0.001	269000
Nursery	0.91	0.96	0.94	0.95	0.97	0.99	0.93	32.05	0.04	764.77
W6a	0.70	0.98	0.90	0.56	0.99	0.93	0.55	188.28	0.44	188.28
Seismic	0.59	0.85	0.70	0.67	0.91	0.92	0.58	7.69	0.37	21.055
Protein	0.51	0.79	0.74	0.60	0.89	0.83	0.49	5.43	0.45	12.02

Table 4. The performance results of the LIBSVM.

Data Set	MCC	Accuracy	Precision	Sensitivity	Specificity	AUC	Uind	*P* +	*P* -	DOR
Adult	0.55	0.85	0.72	0.57	0.93	0.88	0.50	8.42	0.46	18.41
Connect	0.43	0.79	0.79	0.94	0.39	0.87	0.34	1.56	0.14	11.15
Ijcnn1	0.32	0.91	0.69	0.22	0.98	0.99	0.21	49.32	0.79	59.06
Shuttle	0.94	0.98	0.99	0.99	0.95	0.97	0.93	18.61	0.01	1341.7
Nursery	0.90	0. 95	0. 93	0. 94	0. 96	0.99	0. 90	25.21	0.06	392.80
W6a	0.74	0.98	0.90	0.61	0.99	0.95	0.61	242.08	0.39	624.97
Seismic	0.58	0.86	0.71	0.63	0.92	0.89	0.56	8.19	0.38	20.614
Protein	0.55	0.81	0.76	0.63	0.90	0.85	0.53	6.16	0.42	14.84

The results also show that the specificity of NTR-KLR is better than SVM on three data sets and it is the same on three data sets and it is worse only in two data sets. The AUC for NTR-KLR is better than SVM on three data sets, worse on three data sets and it is the same on the other two data sets. The results also show that the Youden's index (Uind) of NTR-KLR is better that SVM in all the data sets except two, which means that NTR-KLR is better than SVM in avoiding the failure. This also shows that Youden's index do not correlate with the standard measures namely accuracy and MCC, which it means that the ability to avoid failure differs from the ability of successful identification of the classification labels. According to the likelihood measure, NTR-KLR is superior overall on shuttle and nurssery data sets, while SVM is superior overall on w6a and protein data set. DOR favors NTR-KLR algorithm on ijcnn1, shuttle and nurssery, while it favors SVM only on w6a data set. The difference in DOR is not significant on the other datasets. In general the superiority of an algorithm is related to the way in which evaluation is measured. The above reported results depict that higher accuracy does not mean overall better performance of an

algorithm. The same conclusion applies to every performance measure if it is considered separately from others. On the other hand, a combination of measures gives a balanced evaluation of the algorithm's performance. To test the statistical difference between the two algorithm overall the various evaluation measures, we used wilcoxon signed rank test. Wilcoxon signed rank test will be appropriate for our data, because it does not assume that the data is homogenous, and normally distributed, also it will not be affected by the size of the data. The results of wilcoxon rank signed test with confidence interval 95% is shown in Table 5 According to the results wilcoxon signed ranks test shows no significant difference between LIBSVM and NTR-KLR, since all the p values are larger than 0.025. This shows that NTR-KLR is as effective as SVM.

Table 5. Wilcoxon signed-ranks test with confidence interval 95%.

Measures	Mcc	Accuracy	Precision	Sensitivity	Specificity	Auc	Uind	P +	P -	Dor
P-Value	0.64	0.61	0.61	0.22	0.34	0.91	0.18	0.58	1.00	0.67

The 95% confidence level accuracies of NTR-KLR on the small size data sets and its comparison with TR-KLR and SVM are shown in Table 6. It is clear that the classification accuracy of NTR-KLR for small size data is comparable to that of SVM and TR-KLR. The only thing is that the ratio of the selected vector m to the number of the data points n will be bigger, but this has no effect on the training time since, n is already small.

Figures 1 and 2 show the ROC curves for adults and seismic data sets respectively. They were constructed using the average estimated probabilities. All of the figures show that the ROC curves for the NTR-KLT is higher than that of the SVM, but at the end of the curve they are almost the same.

Figure 3 shows the execution time of NTR-KLR and SVMs in function of the number of instances. The input data used for the figure is from seismic dataset. It is clear that the execution time of SVM increases rapidly as the number of training instances increases, while the execution time of NTR-KLR increases slowly as the number of instances increases.

Table 6. The 95% confidence level accuracies of NTR-KLR, TR-KLR and SVM.

Data set	NTR-KLR	TR-KLR	SVM
Lonosphere	92.6±2.2	93.7± 2.5	90.6 ±3.1
Diabetes	79.8±2.1	78.0 ±2.9	77.2±3.0
Survival	76.7±4.6	75.4±4.8	75.1±4.9
Liver	69.8±3.5	70.1± 4.8	70.1±4.8
Australian	85.5±2.7	86.1±3.4	85.5±2.3
Heart	82.9±3.6	0.83.8±3.1	0.83.3v3.1

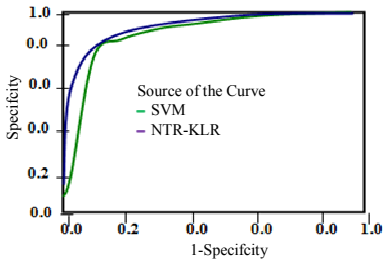

Figure 1. ROC curve for adult data set.

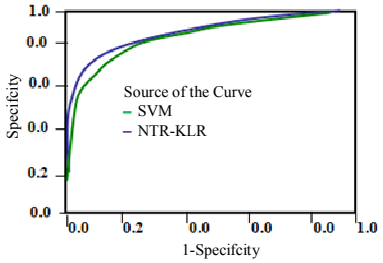

Figure 2. ROC curve for seismic data set.

Figure 3. The execution time in a function of the number of instances for NTR-KLR and LIBSVM.

5. Discussion

Both of the NTR-KLR and LIBSVM perform well in all the data sets; the small differences between the NTR-KLR and LIBSVM results in these data sets were statistically not significant according to Wilcoxon signed-ranks test. To select between NTR-KLR there are four main concerns, these concerns are: First the size of the data set which the classifier is processing, second the need for dealing with input examples of variable length, third the desire to have probabilistic outcomes and fourth the need to perform multiclass classification. When the dataset is very large, people neglect the last two concerns and concentrates on selecting classifier that deal with large datasets effectively. Since, LIBSVM is designed in a way that can handle large scale data sets it becomes the choice for most of the classification purpose. However, LIBSVM does not address the last two concerns directly. KLR is not used in large scale data sets classification although it provides elegant solution to the last two concerns, simply because it is inapplicable in such data sets. NTR-KLR extends the applicability of KLR to be used in large scale data sets. It obtains results that are comparable to the LIBSVM in both small and large scale datasets. This way NTR-KLR can address all of the aforementioned concerns, so it will be a good choice for the classification purpose.

Selecting the number of vectors m from the features matrix using k-means clustering algorithm is an important task in NTR-KLR. A large m can yield high accuracy, but long evaluation time; whereas small m can yield short evaluation time but lower accuracy. To select the optimal value for m a cross validation can be used, by starting with relatively small m and adding more vectors to m until a point where adding more vectors does not improve the accuracy significantly reached.

6. Conclusions

In this paper we have presented a new algorithm based on TR-KLR algorithm, which we called NTR-KLR. NTR-KLR utilizes nystrom method for approximating the eigenvalues and the eigenvectors of the Kernel matrix by selecting $m<<n$ from the features matrix using k-means clustering algorithm. Then the first p eigenvectors from the kernel matrix are used to approximate the eigen decomposition. In this way it is not necessary to store the whole kernel matrix in the memory, but only a $m \times n$ portion of it. We also show that NTR-KLR achieves performance that can be compared with SVM when applied to large-scale datasets e.g., >10,000 using selected vectors of less than or only few hundred. The evaluation time of NTR-KLR is by far less than that of SVM. Moreover, NTR-KLR takes advantage of TR-KLR, which uses unconstrained optimization methods whose algorithms are less complex than those with constrained optimization methods such as SVM.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China under Grant No.61232001, No.61003124, No.61128006, the PhD Programs Foundation of Ministry of Education of China No.20090162120073 and the Freedom Explore Program of Central South University No.201012200124.

References

[1] Canu S. and Smola A., "Kernel Methods and the Exponential Family," *Neurocomputing*, vol. 69, no. 7, pp. 714-720, 2006.

[2] Chang C. and Lin C., "LIBSVM: A Library for SVMs," available at: https://www.csie.ntu.edu.tw/~cjlin/libsvm, last visited 2001.

[3] Cristianini N. and Shawe-Taylor J., *An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge, 2000.

[4] Fan R., Chen P., and Lin C., "Working Set Selection using Second Order Information for Training SVM," available at: http://www.jmlr.org/papers/volume6/fan05a/fan05a.pdf , last visited 2012.

[5] Glas A., Lijmer J., Prins M., Bonsel G., and Bossuyt P., "The Diagnostic Odds Ratio: A Single Indicator of Test Performance," *the Journal of Clinical Epidemiology*, vol. 56, no. 11, pp. 1129-1135, 2003.

[6] Green P., "Iteratively Reweigted Least Squares for Maximum Likelihood Estimation and Some Robust and Resistant Alternatives," *Journal of the Royal Statistical Society, Series B*, vol. 46, no. 2, pp. 149-192, 1984.

[7] Habib M., Hadria I., and Chahira S., "Zernike Moments and SVM for Shape Classification in Very High Resolution Satellite Images," *the International Arab Journal of Information Technology*, vol. 11, no. 1, pp. 43-51, 2014.

[8] Hastie T., Tibshirani R., and Friedman J., *The Elements of Statistical Learning*, Springer, Berlin, 2001.

[9] Hosmer D. and Lemeshow S., *Applied Logistic Regression*, Wiley, London, 2000.

[10] Hsu C., Chang C., and Lin C., "A Practical Guide to Support Vector Classification," *Technical Report*, National Taiwan University, 2010.

[11] Isselbacher K., *Harrison's Principles of Internal Medicine*, McGraw-Hill, USA, 1994.

[12] Jaakkola T. and Haussler D., "Probabilistic Kernel Regression Models," *in Proceedings of Conference on AI and statistics*, Cambridge, UK, pp. 1-9, 1999.

[13] Karsmakers P., "Sparse Kernel-Based Models for Speech Recognition," *PhD Thesis*, Katholieke Universiteit Leuven, Belgium, 2010.

[14] Karsmakers P., Pelckmans K., and Suykens J., "Multi-Class Kernel Logistic Regression: A Fixed-Size Implementation," *in Proceedings of the International Joint Conference on Neural Networks*, FLorida, USA, pp. 1756-1761, 2007.

[15] Koh K., Kim S., and Boyd S., "An Interior-Point Method for Large-Scale l1-regularized Logistic Regression," *the Journal of Machine Learning Research*, vol. 8, no. 8, pp. 1519-1555, 2007.

[16] Komarek P. and Moore A., "Making Logistic Regression a Core Data Mining Tool with TR-IRLS," *in Proceedings of the 5th International Conference on Data Mining*, Washington, USA, pp. 685-688, 2005.

[17] Lin C., Weng R., and Keerthi S., "Trust Region Newton Methods for Large-Scale Logistic Regression," *in Proceedings of the 24th International Conference on Machine Learning*, New York, USA, pp. 561-568, 2007.

[18] Maalouf M., Theodore B., and Adrianto I., "Kernel Logistic Regression using Truncated Newton Method," *Computational Management Science*, vol. 8, no. 4, pp. 415-428, 2010.

[19] Platt J., "Fast Training of Support Vector Machines using Sequential Minimal Optimization," *Advances in Kernel Methods: Support Vector Learning*, 1999.

[20] Suykens J., Gestel T., De Brabanter J., De Moor B., and Vandewalle J., *Least Squares Support Vector Machines*, World Scientific Publishing, Singapore, 2002.

[21] Williams C. and Seeger M., "Using the Nystrom Method to Speed up Kernel Machines," *in Proceedings of the 14th Annual Conference on*

Neural Information Processing Systems, British Columbia, Canada, pp. 682-688, 2001.

[22] Youden W., "Index for Rating Diagnostic Tests," *Cancer*, vol. 3, no. 1, pp. 32-35, 1950.

[23] Zhu J. and Hastie T., "Kernel Logistic Regression and the Import Vector Machine," *the Journal of Computational and Graphical Statistics*, vol. 14, no. 1, pp. 1-8, 2005.

[24] Zhang K., Tsang I., and Kwok J., "Improved Nystrom Lowrank Approximation and Error Analysis," *in Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, pp. 1232-1239, 2008.

Murtada Elbashir received the BSc degree in Computer/statistics from university of Gezira, Sudan, in 2000, The MSc degree in computer information systems from Free State University, Bloemfontein, South Africa, in 2003 and the PhD degree in computer science and technology in Central South University, China, in 2013. His current research interest include: Machine learning and bioinformatics.

Jianxin Wang received the BEng and MEng degrees in computer engineering from Central South University, China, in 1992 and 1996, respectively and the PhD degree in computer science from Central South University, China, in 2001. He is the chair of and a professor in Department of Computer Science, Central South University, China. His current research interests include: Algorithm analysis and optimization, parameraized algorithm, bioinformatics and computer network. He is a senior member of the IEEE.