

Optimal Multi-class Classification with Principal Components

Albert Hoang

Abstract— An approach to build a multi-class classifier is proposed in this paper. This approach consists of a derivation to show under which loss function an optimal classifier can be obtained. It also consists of a method of selection of principal components for multi-class classification through univariate logistic regressions. And it consists of a derivation of certain derivatives to rank the features using two-layered neural network classifier. An experiment of using a two-layered neural network to test the proposed approach was carried out. The performance of the proposed method of data reduction was found better than those of some other methods in this particular experiment. And the features that have high ranking of influence were found credible.

I. INTRODUCTION

Classification or supervised learning concerns with the task of constructing procedures or classifiers based on some known data in order to classify items into appropriate classes. There are two types of classification that could be called uni-class and multi-class. In uni-class or single-label classification an item could belong to exactly one out of many mutually exclusive categories or classes. In multi-class or multi-label classification an item could belong to any number of possibly overlapping categories.

There are a number of issues that need to be addressed in building a multi-class classifier that uses probability for classification. In this paper three issues will be addressed. One is the choice of a loss function for the Bayes rule [3] to apply. Another is how to handle input feature vectors of large dimension. And the last is how to determine the influence each element of the feature vector has on the classification probability. The contribution of this paper is an approach to build a multi-class classifier to address the three issues just mentioned.

In the probabilistic approach to multi-class classification, it is often the joint or marginal classification probabilities are calculated. The Bayes rule is then applied on these probabilities without any specification of what the underlying loss function was [2, 10]. This paper shows that under which loss the joint probability or marginal probabilities should be used to minimize the expected loss.

A method often used in text classification is feature selection that uses contingency tables and chi-square or related statistics [9, 12]. This method has a drawback that it selects each feature independently and ignores the possible joint effect of features. Principal components will be used as a data reduction method in this paper. To select which principal components to be used as inputs to a multi-class classifier, univariate logistic regression will be run on each component, and only components that are significant in these regressions will be selected.

In classification sometimes the ranking of influences that features have in making an item to belong to a specific category is of interest. Methods have been devised to rank the genes in terms of their influences in making a sample to belong to a disease class [5]. In the context of a neural network as a multi-class classifier with principal components as inputs, this paper proposes to assess the influence of each feature by the partial derivative of the classification probability with respect to the feature.

The remaining sections are organized as follows. In section II, derivations are presented to show which multi-class classifiers are optimal under which loss functions. In section III, methods to select principal components using univariate logistic regressions are presented. In section IV, a derivation of the partial derivative of any output of a two-layered neural network with respect to any feature is shown. In section V, an experiment of text classification using the data reduction techniques and ranking of features prescribed in section III and IV is reported. Section VI consists of the conclusions and discussions of this paper.

II. OPTIMAL MULTI-CLASS CLASSIFIERS

In a classification problem, each item is often represented by a pair (\mathbf{x}, k) , where \mathbf{x} is a vector of features of the item, and $k \in \{1, 2, \dots, K\}$ denotes a class out of K mutually exclusive classes the item belongs. This classification is called *uni-class* or *single label* because each item could belong to exactly one class. Let $\Gamma(\mathbf{x})$ be a mapping called *uni-class classifier* that maps each feature vector \mathbf{x} into $\{1, 2, \dots, K\}$ representing K possible classes.

Let $L(k, \Gamma(\mathbf{x})) = I_{\{\Gamma(\mathbf{x}) \neq k\}}$ be a loss function

associated with this classifier $\Gamma(\mathbf{x})$. Here $I_{\{\Gamma(\mathbf{x}) \neq k\}}$ is the indicator function taking value 1 if $\Gamma(\mathbf{x}) \neq k$, taking value 0 otherwise. The average loss, denoted by $B(\Gamma)$, is defined to be the expected value $B(\Gamma) = E_{k,\mathbf{x}}[L(k, \Gamma(\mathbf{x}))]$, which is also called the *Bayes risk*. Here the subscripts denote the expectation with respect to the joint probability of k and \mathbf{x} . Decision theory shows that the Bayes risk $B(\Gamma)$ is minimal if $\Gamma(\mathbf{x})$ is defined as $\Gamma(\mathbf{x}) = \arg \max_k \{\text{Pr ob}(k | \mathbf{x})\}$, where $\text{Pr ob}(k | \mathbf{x})$ is the probability for \mathbf{x} to belong to class k . This result is called the Bayes rule and has been applied in classification [3, 7].

In multi-class or multi-label classification, there are K possibly overlapping classes that items may or may not belong to. Mathematically each item can be represented by a pair (\mathbf{x}, \mathbf{c}) , where \mathbf{x} is a vector of the item's features, and class vector $\mathbf{c} = (c_1, c_2, \dots, c_K)$, where $c_k = 1$ or 0 depending if the item belongs to class k or not respectively. Let $\Gamma: \mathcal{X} \rightarrow \{0,1\}^K$ be a map, called *multi-class classifier* or just *classifier*, mapping each \mathbf{x} in \mathcal{X} to vector

$$\Gamma(\mathbf{x}) = (\Gamma_1(\mathbf{x}), \Gamma_2(\mathbf{x}), \dots, \Gamma_K(\mathbf{x})) \in \{0,1\}^K$$

Here $\Gamma_k(\mathbf{x}) = 1$ or 0 denoting if the item is or is not classified to class k respectively. We see that a multi-class classifier $\Gamma(\mathbf{x})$ is defined if and only if K individual associated uni-class classifier $\Gamma_k(\mathbf{x})$'s are defined. Like in the case of uni-class classifiers, conditions exist for a multi-class classifier $\Gamma(\mathbf{x})$ to have a minimal average loss. Indeed, the derivation of the conditions for the uni-class case can be adapted to the multi-class case as follows.

Given a multi-class classifier $\Gamma(\mathbf{x})$, for any item represented by the pair (\mathbf{x}, \mathbf{c}) , a loss function $L(\mathbf{c}, \Gamma(\mathbf{x}))$ needs to be defined. Two approaches can be used to define this loss function based on the loss functions of the individual uni-class classifier $\Gamma_k(\mathbf{x})$'s, which make up $\Gamma(\mathbf{x})$.

The first approach is to define the loss function as:

$$L^*(\mathbf{c}, \Gamma(\mathbf{x})) = \begin{cases} 1 & \text{if } \Gamma(\mathbf{x}) \neq \mathbf{c} \\ 0 & \text{if } \Gamma(\mathbf{x}) = \mathbf{c} \end{cases} \quad (1)$$

Loss function $L^*(\mathbf{c}, \Gamma(\mathbf{x}))$ thus records no loss when $\Gamma(\mathbf{x})$ is identical to class vector \mathbf{c} , and records the full loss of 1 when $\Gamma(\mathbf{x})$ is not identical to \mathbf{c} . With such a loss function, the multi-class classification problem can be considered as a uni-class problem that has 2^K mutually exclusive uni-class classes $\mathbf{c} = (c_1, c_2, \dots, c_K)$'s, where $c_k = 1$ or 0. The Bayes rule when applied to this uni-class problem guarantees that the classifier Γ^* that minimizes the Bays risk is:

$$\Gamma^*(\mathbf{x}) = \arg \max_{\mathbf{c}} \{\text{Pr ob}(\mathbf{c} | \mathbf{x})\} \quad (2)$$

Where $\text{Pr ob}(\mathbf{c} | \mathbf{x})$ is the probability for \mathbf{x} to belong to uni-class \mathbf{c} .

The second approach is to define the loss as the sum of the loss functions of $\Gamma_k(\mathbf{x})$'s. In this approach, the loss is:

$$L^+(\mathbf{c}, \Gamma(\mathbf{x})) = \sum_{k=1}^K L_k(c_k, \Gamma_k(\mathbf{x})) \quad (3)$$

Loss function $L^+(\mathbf{c}, \Gamma(\mathbf{x}))$ thus records the loss as the number of elements of $\Gamma(\mathbf{x})$ and \mathbf{c} that are not equal. The Bayes risk is defined as:

$$\begin{aligned} B^+(\Gamma) &= E_{k,\mathbf{x}}[L^+(\mathbf{c}, \Gamma(\mathbf{x}))] \\ &= \sum_{k=1}^K B_k(\Gamma_k). \end{aligned}$$

This implies $B^+(\Gamma)$ is minimal at $\Gamma^+(\mathbf{x}) = (\Gamma_1(\mathbf{x}), \Gamma_2(\mathbf{x}), \dots, \Gamma_K(\mathbf{x}))$ if and only if $B_k(\Gamma_k)$ is minimal at $\Gamma_k(\mathbf{x})$ for all $k = 1, 2, \dots, K$. Thus the optimal classifier becomes:

$$\Gamma^+(\mathbf{x}) = \left(\arg \max_{c_k=0,1} \text{Pr ob}(c_k | \mathbf{x}) \right)_{k=1, \dots, K} \quad (4)$$

Where $\text{Pr ob}(c_k | \mathbf{x})$ is the probability for \mathbf{x} to be in class k if $c_k = 1$, not in class k if $c_k = 0$.

According to the above definitions, loss function

L^+ is more flexible than L^* because L^+ can record partial loss, or equivalently partial gain. Loss function L^* on the other hand doesn't record partial loss or partial gain, records just either full loss or full gain. For this reason $\Gamma^+(\mathbf{x})$ should be used when partial loss, or equivalently partial gain, in multi-classification is desired. And $\Gamma^*(\mathbf{x})$ should be used when no partial loss is allowed, just the full gain and full loss.

There are multi-class classifiers proposed in the literature implicitly having used either loss function L^* or L^+ . In [10], classifier $\Gamma^*(\mathbf{x})$ as defined in (2) was constructed by determining the joint probability $\text{Pr ob}(\mathbf{c} | \mathbf{x})$ from the posterior of the parameters involved. For classifier $\Gamma^+(\mathbf{x})$ as defined in (4), there are two ways to construct this classifier since either we can construct K classifiers each calculating one probability $\text{Pr ob}(c_k = 1 | \mathbf{x})$ separately or we can construct one classifier calculating the vector $(\text{Pr ob}(c_k = 1 | \mathbf{x}))_{k=1, \dots, K}$ of all K probabilities at once. In [11], $\Gamma^+(\mathbf{x})$ has been constructed by using K two-layered neural networks, each with just one output calculating $\text{Pr ob}(c_k = 1 | \mathbf{x})$. Classifier $\Gamma^+(\mathbf{x})$ has also been constructed using just one two-layered neural network having K outputs calculating $(\text{Pr ob}(c_k = 1 | \mathbf{x}))_{k=1, \dots, K}$ [2, 11]. The disadvantage of using K neural networks is that the time required for training is long when K is large.

III. CLASSIFICATION WITH PRINCIPAL COMPONENTS

An approach often used in classification of high dimensional data is to reduce the original feature vectors to vectors in lower dimensional space. Many feature selection methods such as information gain and χ^2 reduce the original feature vectors by assessing each feature independently and ignoring the possible correlation among the features. In this section, methods of data reduction with principal components are proposed. Firstly, the original feature vectors will be transformed to vectors of principal components, which are uncorrelated. Secondly, only those principal components that are significant in univariate logistic regressions will be selected as inputs to a chosen classifier. The classifier chosen in this paper for the multi-class classification is a two-layered neural network having K output units, each representing the

probability that an item belongs to a specific class. The neural network is made to become the optimal classifier defined in (4) with \mathbf{x} replaced by the vector of selected principal components.

Let $X_{N \times d}$ be a data matrix consisting of observations $\mathbf{x}_{i \times 1}$'s that are assumed already subtracted by the mean vector. The principal components are defined to be those linear combinations of elements of \mathbf{x} that have maximum sample variance [7]. Principal components can be obtained by applying the singular value decomposition on X to produce:

$$X = UDV' \quad (5)$$

For any \mathbf{x} , $\mathbf{p} = V'\mathbf{x}$ can be shown to be the vector of principal components corresponding to \mathbf{x} [6].

A two-layered neural network with K outputs for multi-class classification can be considered as a regression relating feature vector \mathbf{x} to class vector \mathbf{c} . For every vector \mathbf{x} representing an item or observation, its principal component vector \mathbf{p} also represents the same item, but possibly in a lower dimensional space. The original neural network regression on \mathbf{x} can then be considered as a regression on \mathbf{p} , which can be called *principal component neural network regression*. The problem is then which components should be used as inputs to the network. The method proposed in this paper is to divide principal component vector \mathbf{p} into $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2)$, where \mathbf{p}_1 is independent with \mathbf{c} while \mathbf{p}_2 is not independent with \mathbf{c} , and then use \mathbf{p}_2 as the input to the neural network.

Let p be any principal component, a method to determine whether p is approximately independent to any c_k is to run a univariate logistic regression relating $\text{Pr ob}(c_k = 1 | p)$ to p as:

$$\log \frac{\text{Pr ob}(c_k = 1 | p)}{1 - \text{Pr ob}(c_k = 1 | p)} = \beta_0 + \beta_1 p$$

Under the maximum likelihood method, estimate $\hat{\beta}_1$ of β_1 and the standard error s_1 of $\hat{\beta}_1$ can be obtained through the Newton-Raphson or iterative re-weighted least square method. Also for large sample, when

$\beta_1 = 0$ quantity $z = \frac{\hat{\beta}_1}{s_1}$ asymptotically follows the standard normal distribution, or equivalently z^2 follows the chi-square distribution of one degree of

freedom. Thus we can do a hypothesis testing to test if $\beta_1 = 0$ based on statistic z or z^2 at any significant level α .

Using the above hypothesis testing we can thus determine from the original vector of principal components \mathbf{p} the sub-vector \mathbf{p}_2 that consists of all those p 's that produce significant β_1 's in the corresponding logistic regressions for each of the K classes. This method of determining the reduced set of principal components is called PC_α in this paper.

The reduced set \mathbf{p}_2 can also be determined by following the approaches of taking the weighted sum and maximum scores often used in term selection for text categorization [9, 12] as follows. For every principal component p , and every class k , calculate the value $\chi_k^2(p, k) = z^2$ after each logistic regression run. From the data the probability $\Pr(k)$ that an item belongs to class k can be computed. For the method of weighted sum score, define for every principal component p a weighted score:

$$PC_{avg}(p) = \sum_{k=1}^K \Pr(k) \chi_k^2(p, k)$$

For the method of maximum score, define for every principal component p a max-score:

$$PC_{max}(p) = \max_{k=1, \dots, K} \{\chi_k^2(p, k)\}$$

We then can rank the principal components by either their weighted scores or max-scores, and select those components having the chosen scores greater than a cutoff point to include into \mathbf{p}_2 . These methods are justified by the fact that a component p having high value $\chi_k^2(p, k)$ most likely will have non-zero $\hat{\beta}_1$ in logistic regression for class k , and that value $\chi_k^2(p, k)$ is weighted by the size of class k .

A two-step data reduction has just been proposed. It consists of step 1 of obtaining the principal components from the training data, and step 2 of selecting those principal components that have significant β_1 's in univariate logistic regressions. The selected principal components are then used as inputs to a two-layered neural network classifier that has K output units, each representing $\Pr ob(c_k = 1 | \mathbf{p}_2)$. To make this neural

network an optimal classifier in the form of (4), an item is classified as belonging to any class k iff $\Pr ob(c_k = 1 | \mathbf{p}_2) > \Pr ob(c_k = 0 | \mathbf{p}_2)$, or equivalently $\Pr ob(c_k = 1 | \mathbf{p}_2) > .50$.

IV. RANKING OF FEATURES

For some classification problems, it may be of interest to know which elements of the original vector \mathbf{x} have the largest effects on the probability that \mathbf{x} belongs to a class k , $k = 1, 2, \dots, K$. In the neural network regression considered here, for each k the output is $y_k = \Pr ob(c_k = 1 | \mathbf{p}_2)$, where \mathbf{p}_2 is a vector of selected principal components obtained from one of the methods described in section III. To determine the influence any element x_i of $\mathbf{x} = (x_1, \dots, x_I)$ has on any class k , we can calculate the partial derivative $\frac{\delta y_k}{\delta x_i}$, and rank the effects of x_i 's

based on the averages of these derivatives. A derivation of this derivative is presented below.

Let $\mathbf{X}_{I \times 1}$ be a random feature vector with covariance matrix Σ . By the singular value decomposition we have $\Sigma = W\Lambda W'$, where columns of W are eigenvectors of Σ . From the singular decomposition of X as in (5), each element p_m of vector \mathbf{p}_2 can be expressed as $p_m = \mathbf{v}_m' \mathbf{x}$, where \mathbf{v}_m is the m -th column of matrix V and is also the m -th eigenvector of the sample variance matrix of X [7]. Since each \mathbf{v}_m is asymptotically normally distributed with the corresponding eigenvector of Σ as its mean for large sample [7], \mathbf{v}_m can be approximately considered as constant, unchanged with respect to \mathbf{x} . The partial derivative of p_m with respect to (wrt) the i -th element x_i of \mathbf{x} can be approximated as $\frac{\delta p_m}{\delta x_i} = v_{im}$, which is the i -th element of \mathbf{v}_m .

Consider a two-layered feed forward network with M input units, J hidden units, K output units. A hidden unit h_j and an output unit y_k are of the forms:

$$h_j = \sigma \left(\alpha_{j0} + \sum_{m=1}^M \alpha_{jm} p_m \right)$$

$$y_k = \sigma \left(\beta_{k0} + \sum_{j=1}^J \beta_{kj} h_j \right)$$

where $j = 1, \dots, J$, and $k = 1, \dots, K$. Here M inputs are M selected principal components, and α_{ji} 's and β_{kj} 's are weights, and $\sigma(s) = (1 + e^{-s})^{-1}$. Taking derivative of y_k wrt x_i and using the chain rule, under the assumption that all parameters are constant and approximates of the true parameters, we obtain

$$\begin{aligned} \frac{\delta y_k(\mathbf{x})}{\delta x_i} &= y_k(1 - y_k) \sum_{j=1}^J \beta_{kj} h_j (1 - h_j) \sum_{m=1}^M \alpha_{jm} v_{im} \\ &= \mathbf{v}_i' \boldsymbol{\eta}(\mathbf{x}) \end{aligned}$$

where \mathbf{v}_i is column i -th of matrix V , and $\boldsymbol{\eta}(\mathbf{x}) = (\eta_1(\mathbf{x}), \eta_2(\mathbf{x}), \dots, \eta_M(\mathbf{x}))'$, and

$$\eta_m(\mathbf{x}) = y_k(1 - y_k) \sum_{j=1}^J \alpha_{jm} \beta_{kj} h_j (1 - h_j)$$

Let δ_{ki} be the average of the derivatives of y_k wrt feature x_i across all observations \mathbf{x} 's. When all features x_i 's are in the same scale, the ranking of the influences of all features on y_k for a specific class k can be assessed through the ordering of $\delta_{k1}, \delta_{k2}, \dots, \delta_{kI}$. When $\delta_{ki} > \delta_{kj}$ we can say that the average rate of change of the classification probability for class k due to feature x_i is larger than that due to feature x_j . In a sense feature x_i is more influential than feature x_j in changing the classification probability. Also a ranking of the influences of a specific feature i on all y_k 's can be assessed through the ordering of $\delta_{1i}, \delta_{2i}, \dots, \delta_{Ki}$. When $\delta_{ki} > \delta_{hi}$ we can say that feature x_i is more influential in changing the classification probability for class k than that for class h .

V. EXPERIMENT

An experiment of text categorization has been carried out to test for the feasibility of the data reduction

methods proposed in this paper and to compare the methods' performance with those of other methods. The experiment was also to assess the sensibility of the method of ranking of the influences of features.

A subset of Reuters-21578 corpus consisting of 2,318 news articles already pre-classified into 10 categories was selected for this purpose. Compared to all the categories in the original Reuters-21578 corpus, these 10 categories have the largest number of articles. A news article in the subset considered here could belong to one or more categories. This experiment is a multi-class classification problem.

To build classifier $\Gamma^+(\mathbf{p}_2)$ as defined in (4), where \mathbf{p}_2 is the vector of selected principal components corresponding to any item \mathbf{x} , probability $\text{Prob}(c_k = 1 | \mathbf{p}_2)$ for item \mathbf{x} to belong to any class k needs to be estimated. A two-layered feed forward neural network with 10 outputs was chosen for the estimation of these probabilities. Each output unit k of the neural network was interpreted as the probability for a news article to belong to category k given its feature vector \mathbf{x} . If this probability was greater than .50 then the article would be classified into class k , otherwise classified as not belonging to class k . Such a classification scheme implements the minimum Bayes risk classifier $\Gamma^+(\mathbf{p}_2)$.

The selected set of 2,138 news articles was split into a train data set of 700 articles used for training, and a test data set of 1,618 articles used for testing. A list of stop words, Porter's stemming method, and calculations of global and local weights have been applied to the train data to obtain a total of 5,504 terms and a $700 \times 5,504$ matrix of weights used for training [1]. The same list of stop words, Porter's stemming method, and calculations of local weights were applied to the test data set, but the 5,504 terms and global weights obtained from the train data set were kept the same. After this pre-processing step, the test data set became a matrix of dimension $1,618 \times 5,504$.

To test for the performance of the reduction methods proposed in this paper, the performances of the neural network classifier under these methods and three other reduction methods were compared. The data reduction methods chosen for the comparison purpose here were GSS coefficient, information-gain and chi-square that have been empirically shown to have the best performance [9, 12]. The method of GSS coefficient is denoted by GSS_{Max} , information gain by IG , and the chi-square method with two variations by χ^2_{Max} , and

χ^2_{Avg} . Each of these methods assigns a score to every term to measure the importance of the term in indicating the classes a document belongs. To reduce the number of terms for classification, a threshold is chosen and the terms whose scores are greater the threshold are selected and used as inputs to the neural network classifier.

The train and test data sets were reduced according to the two -step data reduction method proposed in section III. The data matrix X was centered and decomposed by the method of singular value decomposition. The principal components were obtained as the projections of the train data on matrix V 's column space. There were altogether 700 principal components. And there were 242 principal components that have variances of less than .01 and thus could be considered as constant and dropped out from further consideration. A more aggressive reduction on the number of selected principal components, however, was adopted by retaining the first 420 components whose sum of variances is 95% of the sum of variances of the 5,504 terms. Thus after the first step of reduction, 5,504 original terms were reduced to 420 principal components.

In determining which principal components were to be kept for classification out of the 420 components just selected, 420 logistic regressions were run for each of the 10 news categories. In total there were 123 principal components such that each of them has the slope β_1 significant at 5% level in at least one of the 10 news categories. Thus in this second reduction step, called $PC_{.05}$, there were 123 principal components finally chosen as inputs to the neural network classifier. The second data reduction step were also done using the scores PC_{avg} and PC_{max} as defined in section III. Out of the 420 principal components selected from step 1, 123 components having the highest PC_{avg} were also selected to become inputs to the classifier. Another 123 components were selected using PC_{max} . For comparison purpose, term selection methods using the GSS_{Max} , IG, χ^2_{Max} , and χ^2_{Avg} scores were also applied on the 5,504 original terms of the train data set. For each of the methods, the 123 terms that have the largest scores were selected and used as inputs to the neural network classifier.

For testing under the reduction methods proposed in this paper, each item in the test data set was projected to the column space of matrix V obtained from the train data. The projection of each item was a vector of 700 principal components out of which the same 123 components determined from the train data were

retained for testing. For testing under other methods, out of the 5,504 original terms representing an item, the same 123 terms determined from the train data were retained for testing.

All the data reduction methods considered in this experiment were used to reduce the original news articles represented as vectors of 5,504 terms to vectors of 123 elements. The reduced vectors of 123 elements were then used to train and test a neural network classifier of 10 outputs and 62 hidden units under the same training conditions. The performance measures micro and macro F_1 [9, 11] were used to assess how different data reduction methods affect classification. Let r and p be the recall and precision rates of a classifier, multi-class or not, F_1 is defined to be $\frac{2rp}{r+p}$. It can be shown that F_1 is between 0 and 1 and increases if either r or p increases. Since larger r or p indicates better classifier, larger F_1 thus indicates better performance. Also micro F_1 tends to indicate the classifier's performance on common classes, and macro F_1 on rare classes [11].

Under each method of data reduction, the reduced train data was used to train the neural network 10 times to obtain 10 classifiers, each time the micro and macro F_1 were computed for the test data. Also the average and maximum of micro and macro F_1 's were determined from the 10 individual micro and macro F_1 's and considered as the overall performance of the data reduction methods examined. Table I gives the results.

TABLE I
AVERAGE AND MAX MICRO F1, MACRO F1

	Avg Mic F1	Avg Mac F1	Max Mic F1	Max Mac F1
GSS_{Max}	93.01	78.68	93.55	79.86
IG	91.63	82.43	91.86	82.96
χ^2_{Max}	90.24	82.17	90.40	82.56
χ^2_{Avg}	93.71	82.76	94.58	85.45
$PC_{.05}$	94.70	85.66	94.98	86.46
PC_{Max}	94.74	85.72	95.36	86.74
PC_{Avg}	94.60	85.31	94.92	85.96

We can see from the 2nd column of Table 1 that all PC methods have better average micro F_1 than the GSS_{Max} , IG, χ^2_{Max} , and χ^2_{Avg} methods, with the

method PC_{Max} having highest value of 94.74. In fact from the remaining columns 3rd, 4th, and 5th, all PC methods have better average macro F_1 , better max micro F_1 , and better max macro F_1 than the GSS_{Max} , IG, χ^2_{Max} , and χ^2_{Avg} methods. The best method is PC_{Max} , which has the all the measures larger than the measures of all the other methods. PC_{Max} 's measures of average micro, average macro, max micro, and max macro F_1 are 94.74, 85.72, 95.36, and 86.74. The results of this experiment indicate that the two-step data reduction method proposed in this paper has a better performance than other methods for this specific data set and choice of classifier.

With the use of δ_{ki} 's as described in section IV, the ranking of the rates of change of classification probability for each class and due to all the terms was obtained. For each category of news articles, the 3 terms having the highest rates of change for the category are reported here. For the category of articles about *company acquisition*, the 3 terms having the largest change rates are *acquire*, *acquisition*, *company*. These 3 terms can be interpreted as the most influential in affecting the probability for a news article to belong to category *company acquisition*. For the class of articles about *corn*, 3 terms are *corn*, *maiz*, and *feedgrain*. For the *crude oil* category, 3 terms are *oil*, *barrel*, and *crude*. For the *company financial earning* category, 3 terms are *ct (for cents)*, *net*, *loss*. For the *grain* category, 3 terms are *tonn*, *wheat*, *grain*. For the *interest rates* category, 3 terms are *rate*, *day*, *pct (for percentage)*. For the *monetary-foreign exchange* category, 3 terms are *dollar*, *fed*, *drain (like drain reserves)*. For the *ship* category, 3 terms are *port*, *ship*, *gulf*. For the *commercial trade* category, 3 terms are *trade*, *import*, *tariff*. For the *wheat* category, 3 terms are *wheat*, *flour*, *tonn*. The ranking as proposed in section IV indeed picked out the terms that seem influential in making an article belonging to a category.

VI. CONCLUSION AND DISCUSSIONS

In this paper, univariate logistic regression has been used to select principal components, and the selected components have been used as inputs to a two-layered neural network acting as a multi-class classifier. An optimal classification rule has been derived and applied to the neural network to make it an optimal classifier having a flexible loss function. A formula has also been derived to determine the importance of any term to the probability for an item to belong to any class.

The method of partial least squares is another technique of data reduction and has recently been applied to the problem of classification of microarray data [4, 8]. Since partial least square components, like principal components, are linear combinations of elements of feature vectors and are uncorrelated, it is expected that the methods proposed in this paper also work for partial least squares for the multi-class classifications. Specifically univariate logistic regressions can be used to determine the number of partial least square components used as inputs to a two-layered neural network. And the formula used to determine the importance of a feature in the context of principal components could also be used in the context of partial least squares with an appropriate modification and interpretation.

REFERENCES

- [1] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison Wiley, 1999.
- [2] R. A. Calvo, "Classifying Financial News with Neural Networks" *Proc of the 6th Australian Document Computing Symposium*, Coffs Harbor, Australia, Dec. 2001.
- [3] G. Casella and R.L. Berger, *Statistical Inference*. Belmont, CA: Wadsworth, 1993, pp. 123–135.
- [4] P.H. Garthwaite, "An Interpretation of Partial Least Squares", *J. American Statistical Association*, 89(425):122-127, 1994.
- [5] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene Selection for Cancer Classification using Support Vector Machines", *Machine Learning*, 64:389-422, 2002.
- [6] A. Hoang, "Information Retrieval with Principal Components." *Proc. of the International Conference on Information Technology*. Las Vegas 2004.
- [7] R.A. Johnson and D.W. Wichern, *Applied Multivariate Statistical Analysis*, 4th ed. Prentice Hall, 1999.
- [8] D.V. Nguyen and D.M. Rocke, "Tumor Classification by Partial Least Squares using Microarray Gene Expression Data.", *Bioinformatics*, 18:39-50, 2002.
- [9] F. Sebastiani, "Machine Learning in Automated Text Categorization". *ACM Computing Surveys*, Vol. 34, No 1, March 2002, pp.1-47.
- [10] N. Ueda and K. Saito, "Parametric Mixture Models for Multi-class Text". *Neural Information Processing Systems 15 (NIPS 15)*. MIT Press, pp 737-744. 2002
- [11] Y. Yang and X. Liu, "A Re-examination Text Categorization Methods". *Proc. of SIGIR-99*, 22nd ACM International Conference on Research and Development in Information Retrieval. Berkeley, CA 1999.
- [12] Y. Yang and J.O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization". *Proc. of ICML '97*, 14th International Conference on Machine Learning. Nashville, TN, 1997) 412-420