A GPU Accelerated Text Classification Method in E-learning Environment Based on Semi-supervised NMF and SVM

Feng LIU1,*, Liu-tao ZHAO1 and Qian CHEN2

1Beijing Computing Center
2School of Computer Science and Technology, Beijing Institute of Technology
*Corresponding author

Keywords: Text classification, SVM, NMF, CUDA, E-learning.

Abstract. With the beginning of revolution in education E-learning become increasingly popular. It is widespread among people while E-learning platform is used by more people to publish their own resources freely. As a result, a large number of document data need to be processed. Text classification is a key technology to this problem.
In this paper, we propose a frame based on semi-supervised NMF and SVM for text classification. The method of semi-supervised NMF can work out a so called open set problem well. And we also proposed an accelerated version of the algorithm base on CUDA and GPU parallel architecture. These methods can work out the problems well. And finally we test the algorithms in the E-learning environment, than we give the conclusion according to the results.

Introduction

Nowadays, with the development of Internet Technologies, a revolution in education begins and E-learning becomes more and more popular [1]. As to be a new form of education, it is not perfect, and the E-learning itself faces many challenges [2]. Thus, many new techniques need to be developed for E-learning and many new problems need to be solved.
A E-learning platform may be an open platform where people can publish their own materials on the platform freely, it leads to scads of text material accumulating on the platform. Thus organizing such material will be an important issue. Because reasonable recommendation can be made only if all materials were organized and an ideal recommendation can save a lot of time when people finding resources.
Text-classification is an important technique to organize the text materials. The algorithm can classify an article into several classes, thus these documents can be further processed and arranged.
Common text classification algorithms include K-Nearest Neighbor (KNN)[7], Naïve Bayesian classification (NBC), Support Vector Machine (SVM)[4][5], and etc. All of these algorithms have their advantages in different scenarios.
There are two problems you need to solve for the text classification problem. One is feature selection, the other is open set problem.
In this paper, we propose a frame based on semi-supervised NMF and SVM for text classification [3], which can get around these two problems well. We also proposed an accelerated version of the algorithm base on CUDA and GPU parallel architecture [6].

In the rest of this paper, I will firstly introduce some background knowledge, then explain the frame of text classification we proposed, then explain the GPU accelerating version of the algorithm. We test the algorithm with samples in E-learning environment, then show the results. Finally, we give a conclusion. Results and discussion are reported in Section 5. Final conclusions are presented in Section 6.

Related Work

Text classification refers to the process of automatically determining the text category based on the text content under a given classification system.
The research of text classification began earlier. In 1963, Borko and other people proposed the use of factor analysis of the literature in automatic classification [8]. Then many scholars start a fruitful study in this field.

Mmton and Kulm proposed the probability indexing model and applied it to the field of information retrieval [7][9]. 1962 Rosenblatt designed the perception machine, through the threshold of the neurons to deal with two categories problem [10]; Salton in 1975 proposed a vector space model for describing the text[11].

From the 1980s to the 1990s. This stage is mainly the use of traditional knowledge engineering technology, according to the rules of knowledge provided by experts, they manually set up classifiers. This is actually an expert system. Hayes et al. design the CONSTRUE[12], it is a typical representative.

After the 20th century, 90 years. With the development of Internet technology, the rapid increase in the amount of text data, this method cannot meet the needs of practical applications, so it is gradually replaced by the machine learning method. As a result, classification efficiency and accuracy are greatly improved.

At present, the text classification mainly refers to the text classification based on machine learning. This paper also mainly studies the text classification technology based on machine learning. The so-called machine learning refers to the computer to replace people to learn about the world, the transformation of the world's knowledge.

Proposed Method

Text classification is a task that you are given a document, then you need to classify it into one of several predefined categories. There are two problems: feature selection and openset problem. Today these two are still open questions.

For feature selection, you want to extract a set of features such that they are representative and interpretative. All from the community are search effective methods that extract representative and interpretative features. In the next section, we propose using NMF to extract suitable features.

When you train a classifier, the number of classed and class labels are determined. Then the classifier can only classify samples from these categories. When a sample from a new category comes, classifier will misclassify it definitely. This is called “Open set problem”, because the sample set are kind of “open”, it is an opposite term against “close set” machine learning.

Open set problem may arise frequently in E-learning environment, because there more test classes than you can define, and new kind of learning material will appear quickly, the close set classifier may become obsolete soon.

In the next section, we will show how a method called semi-supervised NMF can get around this well.

NMF and Semi-supervised NMF

Firstly, we describe how we convert an article to a single word vector. The first step is you define a dictionary vector, for example [a, an, animal, ..., zig-zag]. Then for each article, you form the input vector w in such way: each entry of the vector is the total count of the relative word in the dictionary in that article. For example, if v = [5, 10, ..., 0]', this means word 'a' appear 5 times in that article, and 'an' 10 times and so on. Then you concatenate several vectors of different articles together [v1, v2, ..., vm] to form the input matrix V. Then this matrix can be used as input matrix V in both NMF and semi-supervised NMF algorithm.

Non-negative matrix factorization (NMF) is a mathematical tool designed to identify a set of non-negative components of an object, it offers dimension reduction and produces useful representations by converting a data matrix to multiplication of two smaller matrices. Unlike similar methods, NMF puts a non-negativity constraint which enables it to form intuitive and parts-based representations of data on its factors. A formal description of nonnegative matrix factorization can be described as follows.
Let us represent m article word vector as an n×m matrix V, where each column, corresponding to an article and m is the number of documents. Based on NMF theory, we can find two new matrices, W and H, to approximate the original matrix $V \approx WH$, or:

$$V_{ij} \approx (WH)_{ij} = \sum_{k=1}^{n} W_{ik} H_{kj}, \quad W \in R^{n \times r}, H \in R^{r \times m} \quad (1)$$

Usually r is chosen to be smaller than r or m, so that W and H are smaller than the original matrix V. In this way, the columns of V are represented as the non-negative linear combination of a collection of basis elements which represent the components of a dataset. This leads to a reduced version of the original face data matrix.

In order to find $V \approx WH$, one of the iterative algorithm for computing NMF with the Euclidean objective function is as follows:

$$H_{kj} \leftarrow H_{kj} \frac{(W^T V)_{kj}}{(W^T WH)_{kj}} \quad (2)$$

$$W_{ik} \leftarrow W_{ik} \frac{(V H^T)_{ik}}{(WH H^T)_{ik}} \quad (3)$$

where the matrices W and H are initialized as nonnegative random matrices, and the updates are done alternatively, until convergence.

In this paper, we propose a method using semi-supervised NMF to extract feature. This method is implemented in two steps: firstly, train the base vectors W1. Then form a new base matrix $W3 = [W1 \ W2]$, where W1 is the one we get from the training step, then use this W1 to process new input samples and just update W2 and keep W1 fixed. When the update is done, you get W2 and H=[H1 ′H2 ′] ′, where W2 is related bases of known classed and H2 is weights of W2. For a single input word vector $w_i$, if H2i is more weighted then H1i, then you can make the decision that this article belong to a new category that you did not define before.

**Support Vector Machine (SVM)**

Support vector machine (SVM) method is based on the principal of maximal margin bound. The aim of SVM is to find one optimal separating hyperplane to separate the two classes of vectors so that the distance from the hyperplane to the closest vectors of both classes is maximized. SVM has achieved much better performance for pattern classification problems by minimizing the Vapnik-Chervonenkis (VC) dimension and achieving a minimal structural risk. It has been recognized as one of the most successful classifier algorithms for many applications including face recognition.

SVM is a very effective binary classifier. Consider n points that belong to two different classes:

$$\{(x_i, y_i)\}_{i=1}^{n} and y_i \in \{-1, +1\} \quad (4)$$

where $x_i \in R^d$ are the training samples and $y_i$ is the class label.

The aim of SVM is to separate these two classes by finding a separating hyperplane:

$$w^T x + b = 0 \quad (5)$$

where $w$ is the normal vector to the hyperplane and b is the corresponding bias term of the hyperplane. The optimal separating hyperplane is the one which separates the training data with the maximum margin.

The SVM optimization objective is defined as follows:

$$\min_{w,b} \frac{1}{2} w^T w$$

subject to the separability constraint:

$$y_i (w^T x_i + b) \geq 1 \quad (7)$$

The solution can be obtained using the Wolfe dual problem with a Lagrangian-multiplier

$$Q(\alpha) = \max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (8)$$
Subject to $a_i \geq 0$ and $\sum_{i=1}^{n} a_i y_i = 0$. Which is easier to solve. The solution is given by

$$w = \sum_{i=1}^{n} a_i y_i x_i, b = -\frac{1}{2} w \cdot [x_r + x_s]$$

(9)

Where $x_r$ and $x_s$ are any two support vectors with $a_r, a_s > 0, y_r = +1$ and $y_s = -1$.

The choice of $C$ is not critical in practice, and we used $C=100$ in all our experiments. In order to classify a new face $x$ with unknown label, the following decision rule is evaluated:

$$f(x) = sgn\left(\sum_{i=1}^{n} a_i y_i (x \cdot x) + b\right)$$

(10)

where the sum runs over all SV $n$ support vectors. Although the linear hyperplane is a natural choice as a boundary to separate classes, it is inadequate to describe the complexity of real article contents. In order to make SVM applicable to nonlinearly separable article data, the kernel based methods is used to map the input data to a feature space by a nonlinear mapping through a kernel function, where inner products in the feature space can be computed by a kernel function without knowing the nonlinear mapping explicitly. To obtain the nonlinear generalization, the linear input space is mapped to a nonlinear kernel feature space:

$$\varphi: R^N \rightarrow F$$

$$x_1 \cdot x_2 \rightarrow \langle \varphi(x_1), \varphi(x_2) \rangle = K(x_1, x_2)$$

(11)

In all the experiments, the Gaussian Kernel function

$$K(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{\sigma}\right)$$

(12)

is used and the parameter $\sigma$ is set $\sigma = 2^m \sigma_0$, where $\sigma_0$ is the standard deviation of the training samples.

CUDA and Parallel Computing on GPU

Graphics processor unit (GPU) to be used for general purpose applications is not an approach that emerged in recent years, but in 2007, with the development of the CUDA architecture, it is expanded rapidly. The CUDA architecture provide to execute general purpose application without knowledge of the graphics processor.

CUDA is a parallel computing platform and application programming interface (API) model created by Nvidia. It allows software developers and software engines to use a CUDA-enabled graphics processing unit (GPU) for general purpose processing – an approach termed GPGPU (General-Purpose computing on Graphics Processing Units). The CUDA platform is a software layer that gives direct access to the GPU’s virtual instruction set and parallel computational elements, for the execution of compute kernels.

GPU Acceleration Method

Writing a CUDA implementation takes a bit more thought. First, the matrices must be copied to GPU memory. Copies between CPU and GPU are relatively slow (ideally 3 GB/s over the PCI bus), and it’s best to avoid them except during initialization or when returning results. This means that in our case it’s better to perform all of the matrix computations on the GPU to avoid extra copies even if certain operations are better suited for the CPU. Element-wise arithmetic is completely data-parallel and is easily accomplished. Other kernels, including the SGEMMs and sums, require a bit of inter-thread communication and are not so trivially parallelized on CUDA.

Luckily, an optimized SGEMM routine is available in the CUBLAS 2.1 library that achieves 60% of theoretical peak performance for large matrices on current GPUs. We can use this routine to do NMF update matrix calculation.
Experiment Design

The whole process consists of two stages, training stage and process stage.

Training Stage
Both NMF bases and SVM need to be trained before you use them.

For a given input matrix $V$, you firstly perform NMF and get $W_1$ and $H$, you keep $W_1$ and use each column to train SVM.

Processing Stage
When you are given a new input matrix $V_3$, you firstly form the base matrix $W_3 = [W_1, W_2]$, keep $W_1$ fixed and only update $W_2$, then you get $W_2$ and $H_3 = [H_1', H_2']$. Then you compare $H_1, H_2$ in each column $H_3$ if $|H_2|/|H_1| > \theta$ the you can label this article as unknown, and do not need to pass it to SVM. Otherwise, pass $H_1$ to SVM, then the whole procedure finishes.

Results

Semi Supervised NMF Performance Results
We test this method in such way: we choose a data set with two classes, then generate some random data samples and treat them as the third unknown class. If we input these random samples directly to the classifier, they will definitely all be wrongly classified. Then we can use this data set to test whether semi-supervised NMF can reduce the error rate.

The chart below shows the test result. The whole bars are organized into 4 groups, which differs in ratios. This ratio is defined as original samples number divided by new random samples number. As the ratio grows, the input data set is contaminated by more ‘wrong’ samples.

As clearly shown in the chart, if the ratio become larger, the performance of method without semi-supervised NMF drops quickly, while the one with semi-supervised machine learning keeps high performance.

Thus we can make a conclusion that semi supervised NMF will increase the system performance when unknown samples are involved.

![Figure 1. Semi supervised NMF error rate.](image)

CUDA Implemented Semi Supervised NMF Algorithm Performance

Now we show the performance results of cuda version of semi supervised NMF training and testing results. Figure 2 and figure 3 show the results. As you can see, cuda version outperforms general version of the algorithm much.

Conclusion

Text classification is now a very popular field of study, and also the most important part of the most important part of machine learning. In this paper, we use semi-supervised NMF and SVM frameworks for text classification. Using semi-supervised NMF to extract features, it works out the problem of feature extraction very well. The proposed algorithm based on CUDA and GPU parallel
architecture has also obtained very good results. We test the two version of the algorithm with sample in E-learning environment, and the results work out well. With the deepening of machine learning technology research, In view of the characteristics of different practical applications and data, especially the Internet data processing and other large-scale complex applications in the data model, the size of the scale and performance bottlenecks and other issues, will become the text classification related research and application of the key and the main breakthrough direction.

![Figure 2. Semi supervised NMF training time with cuda.](image)

![Figure 3. Semi supervised NMF testing time with cuda.](image)

**Acknowledgements**

This paper was funded by oversea talented program of Beijing academy of science and technology. The program number is OTP-2012-015.

**References**


