Pegasos Algorithm for One-Class Support Vector Machine

Changki Lee, Member

SUMMARY Training one-class support vector machines (one-class SVMs) involves solving a quadratic programming (QP) problem. By increasing the number of training samples, solving this QP problem becomes intractable. In this paper, we describe a modified Pegasos algorithm for fast training of one-class SVMs. We show that this algorithm is much faster than the standard one-class SVM without loss of performance in the case of linear kernel.

key words: one-class support vector machine, Pegasos algorithm

1. Introduction

For unsupervised problems, Scholkopf and others proposed one-class support vector machine (one-class SVM) with the goal of estimating the region in the input space where the samples reside [1]. One-class SVMs use only positive examples in training, and they have been successfully applied for novelty/abnormal detection and image retrieval [2]–[4].

One problem of the one-class SVM is the issue of its training, which is a quadratic programming (QP) problem. By increasing the number of training samples, solving this QP problem becomes intractable. Tax and Laskov presented an online training of one-class SVM using a limited sample set [5]. Instead of solving the problem by keeping all data samples as the working set, their method keeps a portion of the data. Tavakkoli and others also used an online training of one-class SVM to model the background of videos [6]. They replace samples in the oldest background frame with the samples belonging to the current background frame. Hua and Ding proposed an incremental learning algorithm for one-class SVM [7]. In this algorithm, the useless sample is discarded and useful information in training samples is accumulated. However, the performances of the online learning algorithms are typically inferior to direct batch optimizers.

In this paper, we propose a modified Pegasos algorithm for fast training of one-class SVMs. The Pegasos algorithm is a stochastic gradient decent method that is originally designed to fit binary classification SVMs [8]. We show that the modified Pegasos algorithm for one-class SVMs is much faster than the standard one-class SVM without loss of performance. Our main contributions are the following:

- We propose a modified Pegasos algorithm for fast training of one-class SVMs.
- We show that the modified Pegasos algorithm for one-class SVM is much faster than the standard one-class SVM training without decreasing the performance.

The rest of this paper is organized as follows. In Sect. 2, we briefly introduce the standard one-class SVMs. Section 3 describes our proposed Pegasos algorithm for one-class SVMs. In Sect. 4, we give the experimental setup and results. The final section gives some concluding remarks.

2. One-Class SVM

We briefly introduce the basic concepts of one-class SVMs in this section. We consider training data $X = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$ and let $\Phi$ be a feature map such that the dot product in the feature space can be computed via kernel $K(x, y) = \Phi(x)^T \Phi(y)$. The standard one-class SVM can be stated as the following objective function to be minimized:

$$
\min_{w, \rho} \frac{1}{2} ||w||^2 + \frac{1}{n} \sum_{i=1}^{n} \xi_i - \rho
$$

subject to $\forall i, w^T \Phi(x_i) \geq \rho - \xi_i, \xi_i \geq 0,$

where $0 < \nu < 1$ is a parameter specified by the user which controls the fraction of outliers over all training samples [1].

After introducing Lagrange multipliers $\alpha_i$ for each vector $x_i$, the dual problem of the optimization problem of Eq. (1) can be obtained as follows.

$$
\max_{\alpha_i} - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j)
$$

subject to $0 \leq \alpha_i \leq \frac{1}{m}, \sum_i \alpha_i = 1$

Since $w = \sum_i \alpha_i \Phi(x_i)$, the obtained hyper-plane $f(x)$, which has maximal distance $\rho/||x||$ from the origin, can be written as

$$
f(x) = w^T \Phi(x) - \rho = \sum_i \alpha_i K(x_i, x) - \rho.
$$

The values of $\alpha_i$ can be determined using the traditional quadratic programming with a linear constraint. The bias term $\rho$ can be also obtained from $f(x_s) = 0$, where $x_s$ denotes one of the support vectors.

Another standard one-class SVM, which is also called support vector data description (SVDD), can be formulated as follows [3].

Manuscript received August 6, 2012.
Manuscript revised January 20, 2013.

The author is with the Department of Computer Science, Kangwon National University, 192–1 Hyoja2-dong Chuncheon-si Gangwon-do, Korea.
a) E-mail: leeck@kangwon.ac.kr (Corresponding author)
DOI: 10.1587/transinf.E96.D.1223

Copyright © 2013 The Institute of Electronics, Information and Communication Engineers
programming, and the values of $R$ can be written as follows.

$$\min_{R,a} R^2 + C \sum_{i=1}^{n} \xi_i$$

$$\text{s.t. } \forall i, \|\Phi(x_i) - a\|^2 \leq R^2 + \xi_i, \xi_i \geq 0,$$  \hspace{1cm} (3)

where vector $a$ denotes the center of the sphere.

After introducing Lagrange multipliers $\alpha_i$ for each vector $x_i$, we can obtain the sphere by solving the following dual problem.

$$\max_{\alpha} \sum_i \alpha_i K(x_i, x_j) - \sum_i \sum_j \alpha_i \alpha_j K(x_i, x_j)$$

$$\text{s.t. } 0 \leq \alpha_i \leq C, \sum_i \alpha_i = 1.$$  \hspace{1cm} (4)

Equation (4) is equivalent to Eq. (2) for the kernel functions satisfying with $K(x_i, x_j) = 1$. The obtained center of the sphere can be written as follows.

$$a = \sum_i \alpha_i \Phi(x_i)$$

The values of $\alpha_i$ can be determined using the quadratic programming, and the values of $R^2$ can be computed from $\|\Phi(x_i) - a\|^2 = R^2$, where $x_i$ denotes one of the support vectors.

3. **Pegasos Algorithm for One-Class SVM**

In this section, we present a modified Pegasos algorithm for one-class SVM. The Pegasos algorithm is a stochastic gradient descent (SGD) method that is originally designed to train binary classification SVMs [8]. The Pegasos algorithm shares the simplicity and speed of online algorithms but is guaranteed to converge to an actual binary SVM solution [8]. Recently Shalev-Shwartz and others proposed many variations of the Pegasos algorithm which include regression, multi-class categorization, and sequence prediction [9]. In this paper, we adapted the Pegasos algorithm for training of one-class SVM. For simplicity of the following theoretical results, we focus on the hyper-planes $f(x) = w^T \Phi(x) - \rho$ with $\rho = 1$ and linear kernel. We reformulate the one-class SVM described in Eq. (1) as follows:

$$\min_{\lambda} \frac{1}{2} ||w||^2 + \frac{1}{n} \sum_{i=1}^{n} l(w; x_i),$$

where $l(w; x_i) = \max(0, 1 - w^T x_i)$. \hspace{1cm} (5)

The one-class SVMs described in Eq. (5) seek to extract a hyper-plane which has the maximal distance $1/||w||$ from the origin because of $\rho = 1$. We replace the objective in Eq. (5) with an approximate objective function,

$$f(w; A_t) = \frac{1}{2} ||w||^2 + \frac{1}{k} \sum_{x \in A_t} l(w; x),$$

Parameter $k$ is the number of examples used for calculating the subgradients, and $A_t$ is the subset of training set $X = \{x_1, \ldots, x_n\}$ ($k = |A_t|$). The subgradient of $f(w; A_t)$ is

$$\nabla f(w; A_t) = \lambda w - \frac{1}{k} \sum_{x \in A_t} x,$$

where $A_t^+$ is the set of examples for which $w$ suffers a non-zero loss. If we choose $A_t = X$ (i.e. $k = n$), then we obtain the subgradient projection method. For the case of $k = 1$, we obtain a variant of stochastic gradient descent method.

A pseudo-code of the modified Pegasos algorithm for one-class SVM optimization problem is depicted in Algorithm 1. The algorithm receives as input two parameters: $T$, the number of iterations to perform; and $k$, the number of examples to use for calculating the subgradients. On iteration $t$, the algorithm first chooses set $A_t$ of size $k$ in $X$ (line 4), finds violated examples in $A_t$ (line 5), and calculates $w_{t+1}$ (line 6).

Algorithm 1. A modified Pegasos algorithm for one-class SVM.

1: Input: $X=(x_1, \ldots, x_n)$, $\lambda$, $T$, $k$
2: Set $w_1 = 0$
3: For $t=1, 2, \ldots, T$
4: Choose $A_t \subseteq X$, where $|A_t| = k$
5: Set $A_t^+ = \{x \in A_t : w_t^T x < 1\}$
6: Set $w_{t+1} = (1 - \frac{1}{t}) w_t + \frac{1}{At} \frac{1}{k} \sum_{x \in A_t^+} x$
7: return $w_T$

One of the main benefits of one-class SVMs is their ability to use non-linear kernels. The common approach for solving the optimization problem for one-class SVM when non-linear kernels are employed is to switch to the dual problem. Following [9], we can directly minimize the primal problem while still using non-linear kernels. At each iteration of the algorithm, $w_{t+1}$ can be written as $w_{t+1} = \frac{1}{t} \sum_{j=1}^{t} \alpha_j \Phi(x_j) = \frac{1}{t} \sum_{j=1}^{t} \alpha_j \Phi(x_j) + \frac{1}{t} \sum_{j=1}^{t} \alpha_j \Phi(x_j)^T \Phi(x_j)$ can be calculated using $\frac{1}{t} \sum_{j=1}^{t} \alpha_j \Phi(x_j)^T \Phi(x_j)$ instead of storing $w_{t+1}$. The computational time in each iteration is $O(n)$ instead of $O(1)$, because of $\frac{1}{t} \sum_{j=1}^{t} \alpha_j \Phi(x_j)^T \Phi(x_j)$.

The pseudo-code of this kernelized implementation of the modified Pegasos for one class SVM is given in algorithm 2.

Algorithm 2. The kernelized modified Pegasos algorithm for one-class SVM.

1: Input: $X=(x_1, \ldots, x_n)$, $\lambda$, $T$, $k$
2: Set $\alpha_1 = 0$
3: For $t=1, 2, \ldots, T$
4: Choose $A_t \subseteq X$, where $|A_t| = k$
5: Set $A_t^+ = \{x_t \in A_t : \frac{1}{t} \sum_{j=1}^{t} \alpha_j \Phi(x_t, x_j) < 1\}$
6: For all $x_t \in A_t^+$, set $\alpha_{t+1}[t] = \alpha_t[t]$
7: For $x_t \notin A_t^+$
8: Set $\alpha_{t+1}[t] = \alpha_t[t] + \frac{1}{k}$
9: return $\alpha_T$
4. Experiments

We implemented the modified Pegasos algorithm for one-class SVM in C++. Our implementation of the modified Pegasos algorithm for one-class SVM is available at http://cs.kangwon.ac.kr/~leeck/software. For comparison, LIBSVM was used as a standard one-class SVM [10]. For all experiments, linear and RBF kernels were used. All experiments were conducted on an Intel Core i7-2600 CPU PC with 3.40 GHz and 16 Giga bytes of RAM. We apply the modified Pegasos algorithm and LIBSVM to real-world data: US postal service OCR (USPS) data set [11] and URL data set [12].

Figure 1 shows plots of the outputs ($w^T \Phi(x)$) on the test set of the USPS data set. The database contains 9298 digit images of size $16 \times 16 = 256$ (the last 2007 constitute the test set). We trained the modified Pegasos algorithm and LIBSVM with the training instances of digit 0 only. Testing was done on both digit 0 (blue) and on all other digits (red). As shown in Fig. 1, the modified Pegasos algorithm performs similar to LIBSVM. However, these two methods have different outputs ($w^T \Phi(x)$) and threshold values ($\rho$). Figure 2 shows Receiver Operating Characteristic (ROC) curves of the modified Pegasos algorithm and LIBSVM with linear and RBF kernels. Table 1 shows the area under the ROC curve (AUC) values of the modified Pegasos and LIBSVM. AUC was calculated using pyroc.py [13]. The ROC curves and AUC values for the modified Pegasos and LIBSVM are very similar (i.e., the performances of two methods are very similar as the decision threshold is varied).

Tables 2 and 3 show experimental results for various values of the modified Pegasos and LIBSVM, respectively. The results show that the modified Pegasos performs similar to LIBSVM, except that these two methods have different $||w||$ and $\rho$.

Figure 3 shows plots of the outputs ($w^T \Phi(x)$) on the test set of the URL data set. The data set consists of 2,396,130 URLs (benign or malicious URLs) and 3,231,961 features (we used first 1,000,000 URLs as a training set and next 10,000 URLs as a test set). We trained the modified Pegasos and LIBSVM with the training instances of benign URLs only. Testing was done on both benign URLs (blue) and malicious URLs (red). From Fig. 3, we can see that the modified Pegasos performs similar to LIBSVM, but these two methods have different outputs ($w^T \Phi(x)$) and threshold values ($\rho$).

Figure 4 and Table 4 show ROC curves and AUC values of the modified Pegasos and LIBSVM with linear and RBF kernels on the test set of the URL data set. The performances of two methods are very similar as the decision threshold is varied.

Figure 5 shows a log-log plot of how CPU-time in-
creases with the size of the training set. Lines in a log-log plot correspond to polynomial growth $O(n^d)$, where $d$ corresponds to the slope of the line. The middle lines show that LIBSVM with linear/RBF kernels and the modified Pegasos with RBF kernel scales roughly $O(n^2)$. The modified Pegasos with linear kernel has much better scaling, $O(n)$.

In overall, we can see that the modified Pegasos algorithm is faster than LIBSVM without decreasing the performance in the case of linear kernel.

5. Conclusion

In this paper, we describe a modified Pegasos algorithm for one-class SVMs. Experimental results showed that this algorithm is much faster than the standard one-class SVM without loss of performance in the case of linear kernel.

Acknowledgments

This study was supported by ‘Development of social web issue detection-monitoring & prediction technology for big data analytic listening platform of web intelligence’ project.

References