Approximate RBF Kernel SVM and Its Applications in Pedestrian Classification

Hui Cao, Takashi Naito, and Yoshiki Ninomiya

Road Environment Recognition Lab, Vehicle Safety Research Cenxter,
Toyota Central R&D LABS.,INC., Nagakute, Aichi 480-1192, Japan.
{caohui, naito, ninomiya}@mosk.tytlabs.co.jp

Abstract. This paper presents an efficient approximation to the nonlinear SVM with Radial Basis Function (RBF) kernel. By employing second-order polynomial approximation to RBF kernel, the derived approximate RBF-kernel SVM classifier can take a compact form by exchanging summation in conventional SVM classification formula, leading to constant low complexity that is only relevant to the dimensions of feature. Experiments on pedestrian classification show that our approximate RBF-kernel SVM achieved classification performance close to the exact implementation with significantly low time and memory.

1 INTRODUCTION

Nonlinear kernel Support Vector Machines (SVM) have shown promising capacities in pattern classification and have been widely used in a number of application areas. The time and memory complexities of classification with nonlinear kernels are $\mathcal{O}(nd)$, where n denotes the number of support vectors and d the feature dimensions. A high-performance SVM classifier will likely need thousands of support vectors, and the resulting high complexity of classification prevents their use in many practical applications (e.g. vehicle-based system) where cost is the critical factor.

Recently linear kernel SVMs have become more and more popular particularly in real-time applications. Due to their dot-product form, linear kernel SVMs are able to be transformed into a compact form by exchanging summation in classification formula, leading to extremely low $\mathcal{O}(d)$ complexity in terms of both time and memory. However, linear kernel SVMs generally yield worse classification performance than that obtained by nonlinear kernel SVMs.

This paper presents an efficient implementation to the nonlinear RBF-kernel SVM. By employing the second-order polynomial approximation of RBF kernel in SVM training, the derived classifier enjoys the favorable classification performance close to that obtained with the exact RBF kernel. but that has a compact form by exchanging summation in classification formula, leading to low complexity $\mathcal{O}(d\frac{d+3}{2})$ that is only relevant to the dimensions of feature.

The rest of the paper is organized as follows: In Section 2 we survey the topic of reducing complexity of nonlinear SVMs. In Section 3 we describe the details

of second-order polynomial approximation to RBF kernel and subsequently the approximate RBF-kernel SVM. In Section 4 we present experimental results on two pedestrian classification datasets. In Section 5 we introduce the extension of approximate RBF-kernel SVM to third-order polynomial approximation. Finally we conclude and present further works in Section 6.

2 Related Works

There are two principle ways to reduce the classification complexity of nonlinear SVMs: (1) reducing the number of support vectors; (2) simplifying the classification process for special kernels.

Several post-processing or direct simplification methods have been proposed for reducing the number of support vectors. After the SVM classifier is built, Burges and Schölkopf [1] apply nonlinear optimization methods to seek sparse representations, while Downs *et al.* [2] present an exact algorithm to prune the support vector set. Lee and Mangasarian [3] select a small random subset from the entire dataset to generate a small rectangular kernel matrix to replace the square kernel matrix used in the nonlinear SVM formulation. Even though these methods (and others not mentioned) can successfully reduce support vectors to some modest size without scarifying the accuracy, the reduced complexities are still much larger than the requirements allowed by many practical applications.

On the other hand, the classification complexity for some special kernels could be significantly reduced by simplifying the classification process. Yang *et al.* [4] use the Fast Gauss Transform to build efficient classifiers based on the Gaussian kernel. Unfortunately, their method is suitable only when the dimension of features is very small. Maji *et al.* [5] recently build an approximate histogram intersection kernel SVM with constant runtime and space requirements by precomputing auxiliary tables. However, their method is suitable only for histogram-like features.

3 Approximate RBF Kernel SVM

Given some training samples $\{y_i, \mathbf{x}_i\}_{i=1}^N$, with the label $y_i \in (-1, +1)$ indicating the class to which the feature vector $\mathbf{x}_i \in \mathrm{R}^d$ belongs. SVM finds linear separating hyperplane with a maximum-margin in the higher feature space induced by kernel function $k(\mathbf{x}, \mathbf{z})$. Some common kernel functions include polynomial, RBF, sigmoid, *etc.*. The normal form of SVM classifier is defined as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i k(\mathbf{x}, \mathbf{v}_i) + b \tag{1}$$

where $\{\mathbf{v}_i\}_{i=1}^n$ are referred to as support vectors which are a small set of training data near the separating hyperplane.

A serious problem with nonlinear kernel SVM is their complexities of classification which are high when a large number of support vectors are needed.

It is desirable to find some ways to enable efficient classification for these nonlinear kernel SVMs. However, it is not easy to find an universal solution for all nonlinear SVMs due to different forms of kernel. In this paper, we are about to implement an efficient classification for RBF kernel, $k(\mathbf{x}, \mathbf{z}) = e^{-\gamma||\mathbf{x}-\mathbf{z}||^2}, \gamma > 0$, which is probably the most popular kernel that succeeded in many areas.

3.1 Approximate RBF Kernel

Assume that feature vector $\mathbf{x}$ is normalized to unit length before training and classifying [1]. The RBF kernel can be written as follows:

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &= e^{-\gamma||\mathbf{x}-\mathbf{z}||^2} \\ &= e^{(-\gamma||\mathbf{x}||^2-\gamma||\mathbf{z}||^2+2\gamma\mathbf{x}^T\mathbf{z})} \\ &= e^{-2\gamma}e^{2\gamma\mathbf{x}^T\mathbf{z}} \end{aligned} \tag{2}$$

In (2), the first term $e^{-2\gamma}$ is constant and thus we may think of RBF kernel as the second term $e^{2\gamma\mathbf{x}^T\mathbf{z}}$ only. In the simplified form of RBF kernel, the value of variable $2\gamma\mathbf{x}^T\mathbf{z}$ is bounded within $[0, 2\gamma]$, as $0 \leq \mathbf{x}^T\mathbf{z} \leq 1$ is always true due to L2-norm normalization. Therefore, if the kernel parameter γ can just take small value, the simplified RBF kernel, $e^{2\gamma\mathbf{x}^T\mathbf{z}}$, is possibly approximated by second-order polynomial approximation, that is

$$k(\mathbf{x}, \mathbf{z}) \approx a + c(2\gamma\mathbf{x}^T\mathbf{z}) + q(2\gamma\mathbf{x}^T\mathbf{z})^2 \tag{3}$$

Below we show three ways of second-order polynomial approximation.

Second-Order Taylor-Expansion Approximation It is known that exponential function can be represented by the Taylor series: $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$. The second-order Taylor expansion $1 + x + \frac{x^2}{2}$ has a reasonable approximation to exponential function if the range of x is less than a small value.

Uniform Second-Order Polynomial Fitting We can also directly fit a second-order polynomial function, $a + cx + qx^2$, to the data points generated from the exponential function e^x, and determine the coefficients a, c, q in a least squares sense. For a certain value that γ takes, a polynomial function is determined so as to best fit the data points uniformly generated by e^x within the range $[0, 2\gamma]$. As a result, we have a set of polynomial functions each corresponding to one of different values that γ could take.

[1] Feature vector normalization is not allowable in all cases, however, in most cases with this operation it could produce same or better classification accuracy.

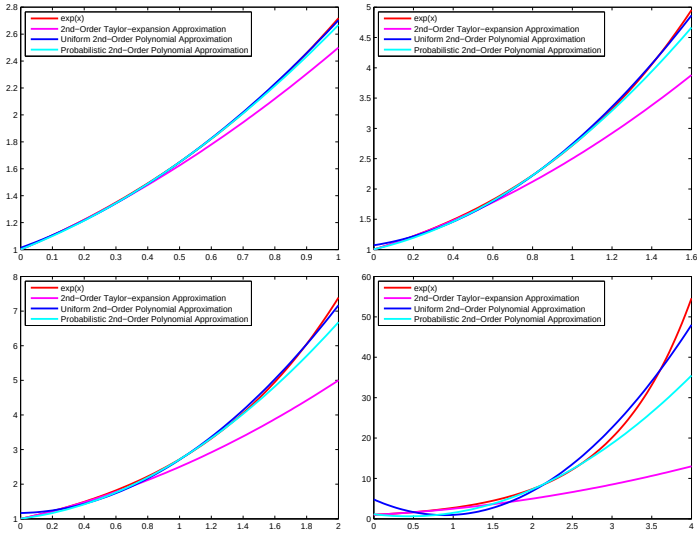

Fig. 1. Comparison of three second-order polynomial approximations to exponential function.

Probabilistic Second-Order Polynomial Fitting The preceding uniform polynomial fitting may be strengthened by taking advantage of the empirical distribution of real-world problems. Data points to be fitted are generated from the exponential function e^x, where $x = \mathbf{x}^T\mathbf{z}$ are sampled according to the empirical distribution of dot-product from training data.

Figure 1 shows comparison of the above-mentioned approximations on different ranges. As the range to be approximated is enlarged, the Taylor expansion approximation becomes more and more biased, whereas the fitting approximations perform consistently well. The probabilistic fitting shows to focus on approximating the middle body of the range in disregard of two ends, because extremely high or low values should generally have low probabilities.

3.2 Efficient SVM Classification with Approximate RBF Kernel

After substituting the approximate RBF kernel (3) into the classification formula (1), we can rearrange the order of computations: the outer summations (between the support vectors) can now be carried out ahead of the inner summations (between the test vector and support vectors). The classification for a test vector can directly employ the results of outer summations calculated offline since the outer summations are unrelated to the test vector. The detail of derivation is given as below:

$$f^2(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i (a + c(2\gamma \mathbf{v}_i^T \mathbf{x}) + q(2\gamma \mathbf{v}_i^T \mathbf{x})^2) + b$$

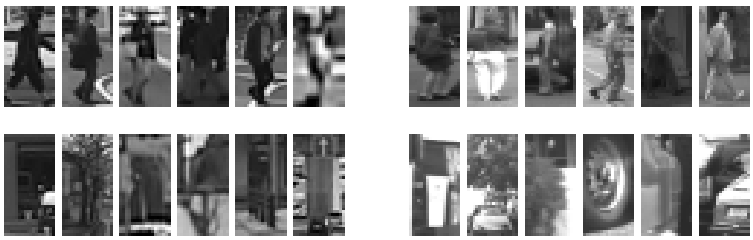

Fig. 2. Left six columns are samples of our own pedestrian dataset and the right six columns are from Daimler-Chrysler dataset (Top: pedestrian, Bottom: non-pedestrian).

$$
\begin{aligned}
&= a\sum_{i=1}^{n}\alpha_i y_i + \sum_{i=1}^{n}\alpha_i y_i (2\gamma c\mathbf{v}_i^T\mathbf{x}) + \sum_{i=1}^{n}\alpha_i y_i 4\gamma^2 q(\mathbf{v}_i^T\mathbf{x})^2 + b \\
&= 0 + \sum_{i=1}^{n} 2\gamma c\alpha_i y_i \sum_{j=1}^{d} v_{ij}x_j + \sum_{i=1}^{n} 4\gamma^2 q\alpha_i y_i (\sum_{j,k=1}^{d} v_{ij}x_j v_{ik}x_k)^2 + b \\
&= \sum_{j=1}^{d}(\sum_{i=1}^{n} 2\gamma c\alpha_i y_i v_{ij})x_j + \sum_{j,k=1}^{d}(\sum_{i=1}^{n} 4\gamma^2 q\alpha_i y_i v_{ij}v_{ik})x_j x_k + b \\
&= \sum_{j=1}^{d}\beta_j x_j + \sum_{j=1}^{d} x_j \sum_{k=1}^{d}\omega_{jk}x_k + b \qquad (4)
\end{aligned}
$$

where $\beta_j = \sum_{i=1}^{n} 2\gamma c\alpha_i y_i v_{ij}$ and $\omega_{jk} = \sum_{i=1}^{n} 4\gamma^2 q\alpha_i y_i v_{ij} v_{ik}$. The first term $a\sum_{i=1}^{n}\alpha_i y_i$ equals zero because $\sum_{i=1}^{n}\alpha_i y_i = 0$.

Due to $\omega_{jk} = \omega_{kj}$, the new form of classifier representation has about $\mathcal{O}(\frac{d(d+3)}{2})$ time and memory complexities. Compared with $\mathcal{O}(nd)$ complexity required by the exact RBF-kernel SVM, the reduction ratio $(\frac{2n}{d+3})$ could be significantly large due to $d \ll n$ in general. For example, if the dimensions of feature is 100 and the number of support vectors is 2000, the reduction of complexity is about 40 times.

4 Experimental Results

We conducted pedestrian classification experiments on two datasets: our own dataset and Daimler-Chrysler Benchmark dataset. Histograms of Oriented Gradients (HOG) [6] is used as features for classification. We tested three types of approximations which are explained in Section 3. Exact RBF-kernel SVM and linear kernel SVM are used for comparison. For convenience, the names of methods are shortened as: approximate SVM, exact SVM and linear SVM. All the experiments were done on an Intel Core2 2.66 GHz machine with 4G RAM. We next describe the experiments on two datasets in detail.

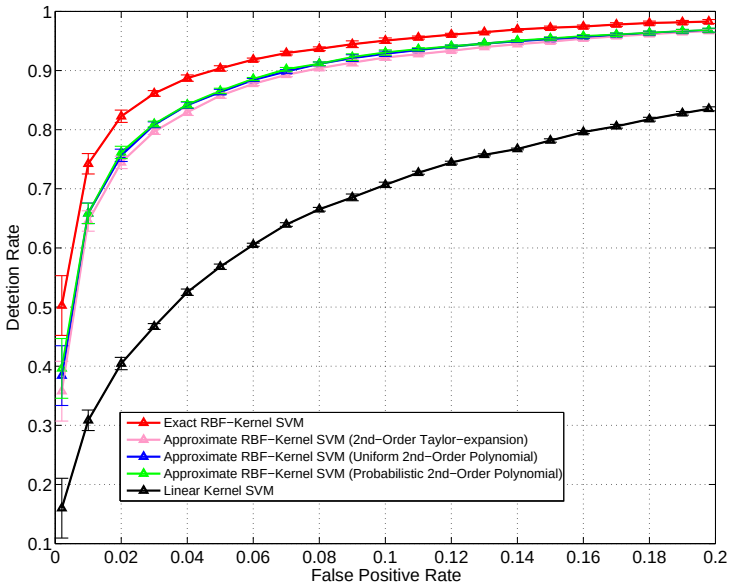

Fig. 3. Performance of our own pedestrian dataset.

4.1 Our Own Pedestrian Dataset

We first use our own pedestrian dataset which is collected with a vehicle-based camera in road environments. The dataset consists of 17,922 pedestrian examples and same amount of non-pedestrian examples. The size of the images is 16×32 pixels. For each image, we compute a 336-dimensional HOG with cell size of 4, block size of 2×2 and histogram bin size of 4. The dataset is randomly divided into three parts: 40% for training, 20% for cross validation and 40% for testing. Classifiers are learned under different parameter settings and the optimal ones are selected via cross validation. Applying the classifiers to test sets yields classification results. In order to reduce the uncertainty in dividing dataset, the above divide-learn-test procedures are repeated for three times. For exact SVM, optimal classifiers are the ones learned with $\gamma = 2$, whereas for two approximate SVMs, optimal classifiers are learned with γ being 0.5 or 0.6. Finally, we obtain for each SVM method 3 ROC curves, from which mean and variance are computed and plotted in Figure 3.

The classification accuracies obtained by the approximate SVMs are about $2 \sim 4\%$ worse than that obtained by the exact SVM. The classification accuracies by the exact SVM classifier that is learned with the same parameter setting as in the approximate SVM, is about 0.5% better than approximate SVMs. As a result, the loss of accuracy may largely due to that approximate SVMs only resemble the sub-optimal exact SVM. Compared three approximations, the

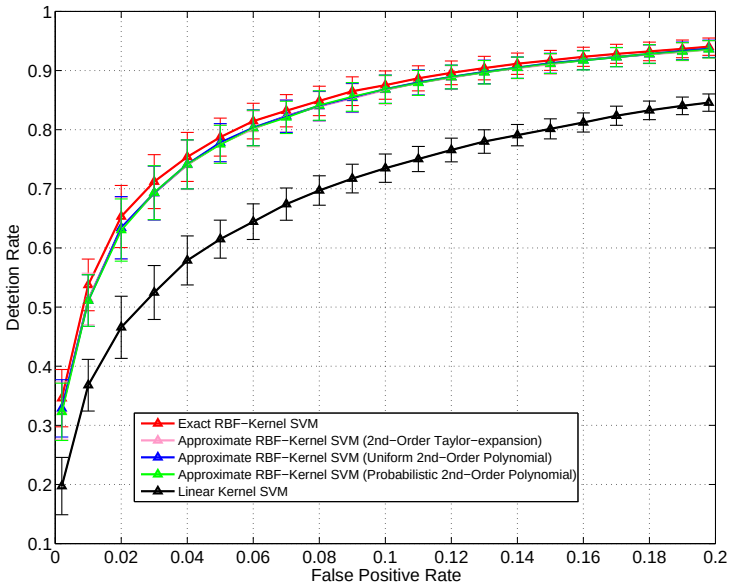

Fig. 4. Performance of Daimler Chrysler pedestrian dataset.

Taylor-expansion approximation is the worst while the probabilistic polynomial approximation shows a bit better than uniform polynomial approximation. The runtime ($0.3ms$) required by the approximate SVMs is 30+ times faster than that of exact SVM ($9.5ms$) and the memory saving is about 40 times. The linear SVM has negligible runtime ($0.0014ms$) and memory, however, the classification accuracy is far below those of nonlinear SVMs.

4.2 Daimler-Chrysler Pedestrian Dataset

We next use the public Daimler-Chrysler Pedestrian Classification Benchmark dataset introduced in [7]. The dataset is divided into five disjoint sets, three for training and two for testing. Each set consists of $4,800$ pedestrian examples and $5,000$ non-pedestrian examples. The size of the images is 18×36 pixels. For each image, we compute a 384-dimensional HOG (with same specification as in previous test). SVM classifiers are generated, each by using two out of the three training data sets and the third training set is used for cross validation for parameter tuning. Applying these classifiers to both test sets yields 6 ROC curves, from which mean and variance are computed and plotted in Figure 4.

Better than the results shown in the previous test, the classification accuracies obtained by three approximate SVMs are very close to the exact SVM. The loss of accuracy is small due to that the value of γ used by exact SVM equals to 0.7,

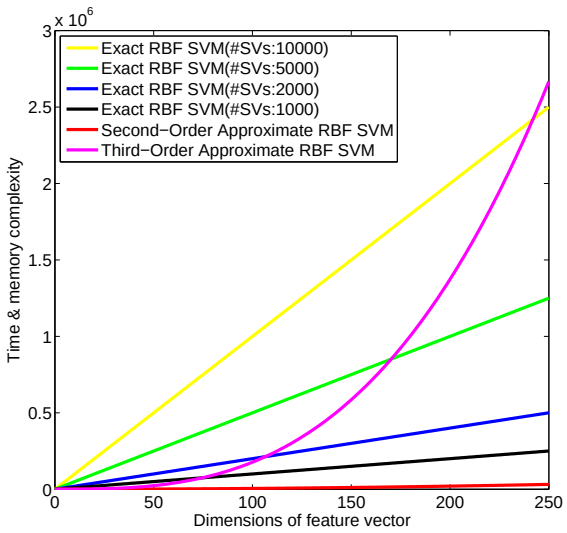

Fig. 5. Classification complexity with regard to number of support vectors and feature dimensions.

which is similar to that used in approximate SVMs, so in this case approximate SVMs can resemble the optimal exact SVM. Three approximations have almost the same classification accuracies. The runtime required by approximate SVMs ($0.38ms$) is over 20 times faster than the exact SVM ($7.7ms$) and the memory saving is about 30 times. The linear SVM shows similar cost and performance as in the previous test.

5 Third-Order Approximation

Although we advocate using the second-order polynomial approximation throughout the paper, the third-order polynomial approximation works too and that would be superior in terms of performance. The third-order approximate RBF kernel has the form,

$$k(\mathbf{x}, \mathbf{z}) \approx a + c(2\gamma \mathbf{x}^T \mathbf{z}) + q(2\gamma \mathbf{x}^T \mathbf{z})^2 + h(2\gamma \mathbf{x}^T \mathbf{z})^3 \tag{5}$$

substituting which into the classification formula (1) leads to the third-order approximate RBF-kernel SVM as follows:

$$f^3(\mathbf{x}) = \sum_{j=1}^{d} \beta_j x_j + \sum_{j=1}^{d} x_j \sum_{k=1}^{d} \omega_{jk} x_k + \sum_{j=1}^{d} x_j \sum_{k=1}^{d} x_k \sum_{s=1}^{d} \theta_{jks} x_s + b \tag{6}$$

where β_j and ω_{jk} are the same coefficients as in the second-order approximation, and $\theta_{jks} = \sum_{i=1}^{n} 8\gamma^3 h \alpha_i y_i v_{ij} v_{ik} v_{is}$.

The new form of classifier representation needs $\mathcal{O}(\frac{d(d^2+6n+11)}{6})$ time and memory complexities. Figure 5 shows analysis of time and memory complexities for RBF-kernel SVM with the second-order and third-order approximations, with respect to various feature dimensions and number of support vectors. It shows that the third-order approximation is efficient only when the dimensions of feature is small.

6 Conclusions

In this paper we present an efficient SVM classifier with the approximate RBF kernel based on low-order polynomial approximation. Experimental results on two pedestrian classification datasets show that the approximate RBF-kernel SVM achieved classification performance comparable to the exact implementation, with significantly reduced complexity in terms of both runtime and memory. Our approximate method would be a better balanced choice from cost and performance perspectives and be particularly suited for the cases with small to medium features and lots of support vectors.

One of our further works is to improve classification performance of the approximate RBF-kernel SVM. Since the complexity of our method is independent of the number of support vectors, we would like to investigate how much the classification performance can be improved by (1) seeking better second-order approximation; 2) collecting many more support vectors through learning from large data sets. In addition, we want to apply the third-order polynomial approximation to problems with small number of features (say, several dozens).

References

1. C. J. C. Burges and B. SchNolkopf: Improving the accuracy and speed of support vector learning machines. NIPS, 375-381 (1997)
2. T. Downs, K. E. Gates, and A. Masters: Exact simplification of support vector solutions. Journal of Machine Learning Research, 2:293-297 (2001)
3. Y.-J. Lee and O. L. Mangasarian: RSVM: Reduced Support Vector Machines. First SIAM International Conference on Data Mining, 350-366 (2001)
4. C. Yang, R. Duraiswami and L. Davis: Efficient Kernel Machines Using the Improved Fast Gauss Transform. NIPS (2004)
5. S. Maji and A. C. Berg and J. Malik: Classification Using Intersection Kernel Support Vector Machines is efficient. IEEE CVPR (2008)
6. N. Dalal and B. Triggs: Histograms of Oriented Gradients for Human Detection. CVPR, 886-893 (2005)
7. S. Munder and D. M. Gavrila: An Experimental Study on Pedestrian Classification. IEEE T. PAMI, 28(11):1863-1868 (2006)