Solving Nonlinear SVM in Linear Time? A Nyström Approximated SVM with Applications to Image Classification*

Ming-Hen Tsai
Dept. of Computer Science
Columbia University, USA
mt2767@columbia.edu

Yi-Ren Yeh
Intel-NTU Connected Context Comp. Center
National Taiwan University, Taiwan
yryeh@ntu.edu.tw

Yuh-Jye Lee
Dept. of CSIE
National Taiwan Univ. Science & Tech., Taiwan
yuh-jye@mail.ntust.edu.tw

Yu-Chiang Frank Wang
Research Center for IT Innovation
Academia Sinica, Taiwan
ycwang@citi.sinica.edu.tw

Abstract

In this paper, we improve the efficiency of kernelized support vector machine (SVM) for image classification using linearized kernel data representation. Inspired by Nyström approximation, we propose a decomposition technique for converting the kernel data matrix into an approximated primal form. This allows us to apply the approximated kernelized data in the primal form of linear SVMs, and achieve comparable recognition performance as nonlinear SVMs do. Several benefits can be observed for our proposed method. First, we advance basis matrix selection for decomposing our proposed approximation, which can be viewed as feature/instance selection with performance guarantees. More importantly, the proposed selection technique significantly reduces the computation complexity for both training and testing. Therefore, the resulting computation time is comparable to that of linear SVMs. Experiments on two benchmark image datasets will support the use of our approach for solving the tasks of image classification.

1 Introduction

Support vector machines (SVM) have been widely applied for solving machine learning tasks due to its generalization. Using kernel functions, SVM can be learned with its dual form in a higher order feature space and solves classification problems which are linearly non-separable. However, construction of kernel matrices in an SVM dual form is computationally expensive, and the size of the kernel matrix is super-linearly proportional to that of the dataset. Therefore, it is typically undesirable (or even infeasible) to train nonlinear SVMs for large-scale problems. Inspired by the recent success of LIBLINEAR [1] and PEGASOS [2], which solve linear SVMs with a fast convergence guarantee, researchers have been utilizing the linear SVM models for solving nonlinear problems for solving larger-scale problems. For example, exact explicit feature mapping [3], approximated explicit feature mapping [4], and projection methods [5] are among the representative techniques.

Among the above approaches, projection methods are commonly used since they need not determine nonlinear feature mapping explicitly, and one can directly apply the kernelized data as inputs to learn linear SVMs. For example, the generalized support vector machine (GSVM) [6] takes the full kernel matrix as a form of kernelized input data for solving linear SVM. Since its computational load depends on the number of columns in the kernel matrix, the reduced SVM (RSVM) [7] is proposed to construct a reduced kernel matrix for decreasing the computation complexity. However, the performance of this type of approaches will be sensitive to the derived projection bases, and how to select a proper set of such bases is typically a challenging task.

In this paper, we investigate a well-known kernel approximation technique, *Nyström approximation*. We propose to decompose a kernelized data matrix with *Nyström* approximation for generating lower dimensional data, and thus one can apply linear SVMs in solving larger-scale problems. We present a feature/instance selection strategy for approximating the kernelized data by selecting a compact subset from the original kernel data matrix. The proposed technique allows us to take the advantages of computation efficiency of linear SVM models, while preserving the classification capability of the nonlinear ones. As verified later by our experiments, our proposed SVM is able to achieve comparable performance as nonlinear SVMs do, while the computation complexity for both training and test is remarkably reduced and comparable to that of linear SVMs.

2 Nyström Approximation for SVM

We first briefly review the SVM in two equivalent formulations, i.e., primal and dual forms. Advantages and challenges using the dual form will be discussed, and we present the existing Nyström approximation for solving SVM on large-scale problems. Note that we assume there exist ℓ training instances in a d-dimensional space, i.e., $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_\ell]^T \in \boldsymbol{R}^{\ell \times d}$ with labels $\boldsymbol{y} = [y_1, y_2, \dots, y_\ell]^T$.

2.1 SVM in Primal and Dual Spaces

The primal-form support vector machine (SVM) learns an optimal separating hyperplane to distinguish data between two classes, and it solves the following

[*] A longer version of this paper (including code) is available at http://github.com/scan33scan33/kernel-decomposition.

optimization problem:

$$\min_{\boldsymbol{w},b} R(\boldsymbol{w}) + C\sum_{i=1}^{\ell} \xi(\boldsymbol{w}; \phi(\boldsymbol{x}_i), y_i). \qquad (1)$$

$R(\boldsymbol{w})$ is the regularization term (e.g., $R(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$), $\xi(\cdot)$ is the loss function, and C is the trade-off between $R(\boldsymbol{w})$ and $\xi(\cdot)$.

We note that ϕ in (1) maps $\boldsymbol{w}$ and $\boldsymbol{x}_i$ into a higher order feature space, which exhibits improved classification capability. Since solving (1) in a high-dimensional space is typically not tractable, one can apply a proper kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$ and derive $Q = K(X, X) = \{K(\boldsymbol{x}_i, \boldsymbol{x}_j)\}_{ij}$ as an $\ell \times \ell$ kernel matrix. As a result, without the need to calculate ϕ explicitly, one can convert (1) into the following dual form, which can be easily solved using quadratic programming techniques:

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \boldsymbol{e}^T\boldsymbol{\alpha} \\ &\text{s.t. } \boldsymbol{\alpha}^T\boldsymbol{y} = 0 \text{ and } 0 \leq \alpha_i \leq C \quad \forall i. \end{aligned} \qquad (2)$$

2.2 Nyström Approximation for Large-Scale SVM

Although the use of mapping functions and kernels provides improved classification ability, computation of a kernel matrix takes $O(\ell^2 d)$ and thus is very computationally expensive especially for large-scale problems (i.e., ℓ is large). Among low-rank approximation techniques to alleviate this problem, we advance Nyström approximation [8] due to its simplicity in decomposing the bases of the kernel matrix.

We now discuss the use of Nyström approximation for large-scale SVMs. Suppose that we require a rank-$\tilde{d}$ approximation of the kernel matrix $Q = K(X, X)$, and we assume $\tilde{d} \ll l$ as in common low-rank approximation settings. Nyström approximation takes the form $\tilde{Q} = PW_{\tilde{d}}P^T$, where we have $P = K(X, B) \in \boldsymbol{R}^{\tilde{\ell}\times d}$, and the matrix $B \in \boldsymbol{R}^{\tilde{d}\times d}$ consists of $\tilde{d}$ landmark examples which can be viewed as *projection bases*. Here, we use $W_{\tilde{d}} = W^{-1}$, where $W = K(B, B)$ as proposed in [8]. Alternatively, as pointed out in [7], one can use $W_{\tilde{d}} = I$ as an aggressive approximation. In this case, the computation time for P and W^{-1} are $O(\ell\tilde{d}d)$ and $O(\tilde{d}^3)$, respectively. Evaluation of each element takes $O(\tilde{d})$ time. It is worth noting that, although the computation time is reduced remarkably, it may be still proportional to ℓ^2 if all kernel elements need to be computed. To further reduce the computational cost, we utilize the Nyström Approximation for solving SVM in the primal, which decreases the computation time to $O(\ell d\tilde{d})$ (as detailed in Section 3).

3 Solving the Approximated Dual Problems in the Primal

For large-scale classification problems, it is always desirable to design classifiers with performance guarantees without remarkably sacrificing computation or memory costs (even training can be done off-line). Although the primal linear SVM model is able to perform both training and testing in linear time, nonlinear SVMs often exhibit improved capability in solving more challenging tasks. To address the above problems, we propose to use L1-regularized SVM and derive a compact set of Nyström approximated bases for nonlinear SVMs. As detailed later, the proposed method will be linear in both selecting the basis and constructing the primal formulation. More importantly, it will be sub-linear in performing subsequent training and testing processes, and thus achieves comparable performance as nonlinear SVMs do.

Algorithm 1 Nyström-Approximated Primal SVM

```
Input  Training data with labels $(\boldsymbol{y}, X)$, kernel function
       $K$, and the expected basis size $s$.
Predefined  A basis selection algorithm $\mathcal{A}(\boldsymbol{y}, X, K, s)$.
1. Get $B \leftarrow \mathcal{A}(\boldsymbol{y}, X, K, s)$
2. Compute $R \leftarrow$ Cholesky decomposition of $K(B, B)$
3. Let $f(X) \equiv K(X, B)R^{-1}$ be the hash function to do
   the data transform
Return $f(X)$
```

3.1 Scaling Up : Nyström Approximation in the Primal

In this paper, we aim at utilizing a linear SVM model to address nonlinear problems via decomposing Nyström approximation. With $\phi(\boldsymbol{x}) = \boldsymbol{x}$ and no parameter b in (1), we have a linear SVM model in which the kernel data matrix X satisfies $K(X, X) = XX^T$. When applying Nyström approximation, the approximated kernel $\tilde{Q}$ (see Section 2.2) can be decomposed into $\bar{X}\bar{X}^T = \{\bar{\boldsymbol{x}_i}^T\bar{\boldsymbol{x}_j}\}_{ij}$. Thus, solving a nonlinear SVM with $\tilde{Q}$ can be viewed as learning a linear SVM model with data $\bar{X}$. To construct $\bar{X}$, we have:

$$\tilde{Q} = PW^{-1}P^T = P(R^TR)^{-1}P^T = PR^{-1}(PR^{-1})^T, \qquad (3)$$

where $R^TR = W + \sigma I$ is derived by Cholesky decomposition with a small positive σ (for the full-rank guarantee). With $\bar{X} = PR^{-1} \in \boldsymbol{R}^{\ell\times\tilde{d}}$, we obtain the primal-form data as desired.

Although the above decomposition technique allows one to solve nonlinear SVMs with linear models, selecting a preferable basis matrix will *not* be trivial. More specifically, even using the kernelized data in its primal form, it still requires high computational costs in constructing the kernel matrix and deriving the optimal basis matrix for such a decomposition. Therefore, we need a fast and effective way to choose a compact and representative basis matrix to make our algorithm feasible, and thus this decomposition technique would be preferable for large-scale problems. Algorithm 1 shows a general framework for performing basis selection, which utilizes a hash function to transform data to an approximated non-linear space.

3.2 Random Basis Selection for Nyström Approximation

A simple and straightforward way to address the above concern is to randomly sample instances from the training set as the basis matrix. When utilizing the matrix $W = K(B, B)$, a PAC-style theorem has

Algorithm 2 Basis Selection Algorithm $\mathcal{A}$

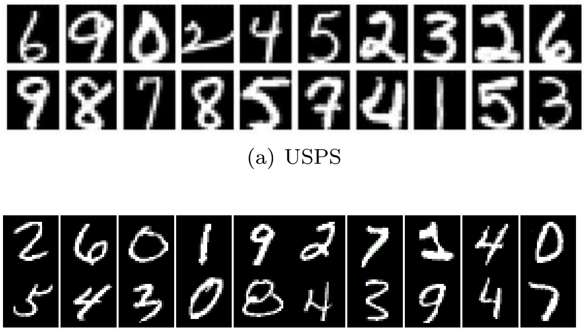
```
Input Training data with labels $(\boldsymbol{y}, X)$, kernel function
    $K$, and the expected basis size $s$
1. Get a initial set of basis $B^{initial}$ with size $\sqrt{\ell}$ from $X$.
   Select a total of $\sqrt{\ell}$ instances (by stratified random
   sampling) from each class as $(\boldsymbol{y}^{subset}, X^{subset})$.
2. $X^{select} \leftarrow K(X^{subset}, B^{initial})$.
3. Train L1-regularized SVM on $(\boldsymbol{y}^{subset}, X^{select})$ and a
   sparse solution $\boldsymbol{w}$ for each class.
4. $B \leftarrow$ columns in $B^{initial}$ where the corresponding ele-
   ment in $\boldsymbol{w}$ is non-zero.
5. If $s$ is defined, $B \leftarrow s$ centroids generated by k-means
   of $B$ (as proposed in [9]).
Return $B$
```

been provided in [5] suggesting that only $O(1/\epsilon)$ instances are required in order to achieve ϵ error in ideal cases. The effectiveness of such a random sampling strategy has been verified in reduced support vector machine (RSVM) from a statistical point of view [7]. In practice, a stratified random sampling is applied, which randomly samples subsets with the same size from each class to alleviate unbalanced data learning problems. In practice, however, it is not clear how to determine the optimal size when sampling such random basis matrices. Performing cross-validation will not be preferable for large-scale problems, since it is still very time consuming even with the use of primal-form kernelized data and linear SVM models.

3.3 Informative Basis Selection for Nyström Approximated SVM

Recall that, in Section 2.2, we have the j-th column of $\bar{X}$ generated by the j-th column of B when $W = I$. Therefore, finding a basis set that maximizes the accuracy of the approximated model on X will be equivalent to performing feature selection on $\bar{X}$. In our work, we have $R(\boldsymbol{w}) = \|\boldsymbol{w}\|_1$ in (1), and we solve this L1-regularized SVM for obtaining a sparse solution $\boldsymbol{w}$. The reason why we choose this selection technique is that it shares the same loss as the SVM objective function does. Moreover, it is able to identify a sparse subset of features/instances with complexity $O(\ell d) \times \#iterations$, which is linear to the dataset size.

However, solving the L1-regularized optimization problem can be intractable. Inspired by the advantages of randomness utilized in [5, 7], we start from a randomly chosen subset, followed by the above L1-regularized SVM for basis selection. Let B be the candidate basis vector set with size ℓ_b. A naive way for designing $\mathcal{A}$ is to generate the primal data by $K(X, B)$. It takes $O(\ell_b \ell d)$ to compute such a kernel matrix. However, Its computational cost is still high when ℓ is large. To alleviate this problem, we choose a sampled subset with size ℓ_s and limit the size of the candidate basis ℓ_b when computing the primal data during the selection process. We further enforce $\ell_s \ell_b \sim \ell$ in our implementation. As a result, the kernel computation time turns into $O(\ell_b \ell_s d) = O(\ell d)$. It can be seen that the resulting computation time is now linearly proportional to the size of the sampled subset. In our framework, we choose $\ell_s = \sqrt{\ell}$ and $\ell_b = \sqrt{\ell}$. Our proposed basis selection algorithm is shown in Algorithm 2.


(a) USPS

(b) MNIST

Figure 1. Example images from the USPS and MNIST handwritten digit datasets.

Table 1. Dataset descriptions (with the number ℓ of instances and the dimension d of the data). The sizes for storing the data ℓd and the associated kernel matrices ℓ^2 are also listed.

	ℓ	d	kernel size ℓ^2	data size ℓd
USPS	7291	256	53M	2M
MNIST	60000	780	3.6G	47M

We note that, in Algorithm 2, we first apply the same stratified sampling strategy when selecting the candidate instances for boosting classification performance. Since our proposed method only requires a subset of instances (and the associated class labels) for learning the compact data representation, our framework can be extended to semi-supervised or privacy preserving classification problems. Furthermore, our setting can also be easily extended for learning models with multiple kernels. In such cases, the selection of representative bases using L1-regularized SVMs (i.e., Algorithm 2) can be performed on all random bases generated by different base kernels.

4 Experiments

4.1 Experimental Settings

We conduct experiments on two benchmark image datasets: USPS and MNIST. Table 1 describes these two datasets, including the sizes of the data and those of the corresponding kernel matrices. It can be seen that, when the dataset is large (e.g., MNIST), computing and storing the entire kernel matrix for nonlinear SVMs will be very expensive.

For experiments, we randomly split 70% of the data for training and the remaining 30% for testing. We repeat this process five times and report the average recognition rates with their standard deviations. Each instance is scaled to unit-length ($\|\boldsymbol{x}_i\|_2 = 1$), since this is observed to produce better performance.

4.2 Discussions

In our experiments, we consider the use of RBF kernels for nonlinear SVM classification. We compare the performance and computation time of our Nyström approximated SVM in primal form to those produced by standard linear and nonlinear SVMs. We perform a

Table 2. Comparisons of accuracy and computation time on the two datasets. Training time (in seconds) is computed for the entire data set, while testing time is the average time computed for each instance (in milliseconds).

USPS	Acc.	Training Time	Test Time
Ours	97.057	3.764 (s)	1.486 (ms)
RBF SVM	98.007	14.507 (s)	1.888 (ms)
Linear SVM	95.009	2.274 (s)	0.036 (ms)
MNIST	Acc.	Training Time	Test Time
Ours	95.983	57.318 (s)	0.578 (ms)
RBF SVM	98.547	3650.334 (s)	29.138 (ms)
Linear SVM	91.917	14.510 (s)	0.156 (ms)

five-fold cross validation to select the parameters γ and C, and the search spaces are $\log_2 \gamma \in \{-7, -5, \dots, 1\}$ and $\log_2 C \in \{-3, -1, \dots, 15\}$, respectively.

Table 2 shows the experimental results for the three SVM models. As observed, our method achieved improved recognition rates than linear SVMs, while the time for training and testing using our proposed model is comparable to that of linear SVMs. On the other hand, although the use of nonlinear SVMs reported a slightly better recognition performance than ours, it required remarkably longer time for both training (e.g. 3650.3 vs. 57.3 seconds) and testing (e.g. 29.1 vs. 0.5 seconds). This is because that the standard nonlinear SVM utilizes the full kernel matrix, and its computation time will be quadratically scaled-up with ℓ. It is worth noting that, training and test time reported in Table 2 involves feature hashing and deriving the SVM model/ouput. If needed, hashing features can be performed off-line.

4.3 Comparisons to Random Basis Selection

To verify the effectiveness of our proposed basis selection approach, we compare the performance using the basis matrix determined by our method with that derived by the random sampling strategy (as the RSVM did [7]). We plot the recognition rates using these two different methods on the two datasets in Figure 2, in which the horizontal axis shows the number of selected bases, and the vertical axis denotes the recognition rates. From this figure, it can be seen that our selection method yielded higher accuracy than random sampling did given the same basis size. This confirms that we are able to identify a better (i.e., more informative) basis matrix for designing our Nyström approximated SVM.

5 Conclusion

We proposed a method to solve an approximated dual SVM in the primal based on Nyström approximation. Without the need to store the entire kernel data matrix like nonlinear SVM does, our approach automatically determines the optimal basis matrix for decomposing Nyström approximated kernel data, which allows us to solve linear SVMs with complexity only linearly scaling up with the dataset size. Experiments on two image datasets confirmed that our proposed model achieved improved recognition accuracy than the standard linear SVM did and with comparable computation time. Training and testing time using our proposed was remarkably reduced compared to that of nonlinear SVMs. Future research directions include the extension of our proposed model for a multiple kernel setting, possible integration of unsupervised feature selection approaches, and our proposed model for semi-supervised or unsupervised learning problems.

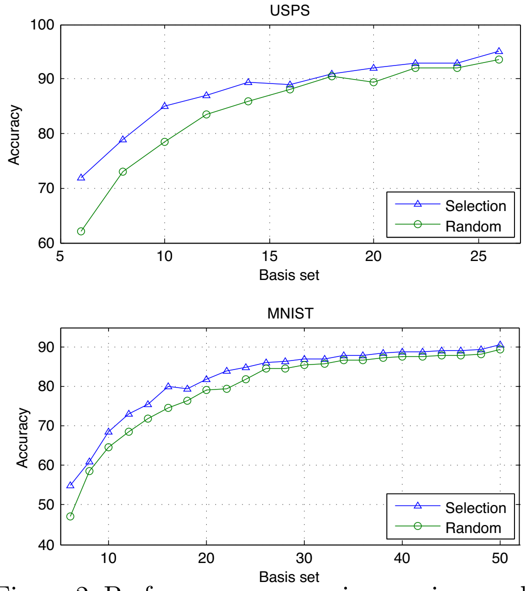


Figure 2. Performance comparisons using our basis selection method (in blue) vs. random sampling (in green) on USPS and MNIST datasets.

Acknowledgement

This work is supported in part by National Science Council of Taiwan via NSC100-2221-E-001-018-MY2 and NSC101-2218-E-011-006.

References

[1] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *JMLR*, 2008.

[2] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: primal estimated sub-gradient solver for SVM," *ICML*, 2007.

[3] Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin, "Training and testing low-degree polynomial data mappings via linear SVM," *JMLR*, 2010.

[4] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE CVPR*, 2010.

[5] M.-F. Balcan, A. Blum, and S. Vempala, "Kernels as features: On kernels, margins, and low-dimensional mappings," *Machine Learning*, 2006.

[6] O. L. Mangasarian, "Generalized support vector machines," *Advances in Large Margin Classifiers*, 2000.

[7] Y.-J. Lee and S.-Y. Huang, "Reduced support vector machines: A statistical theory.," *IEEE Trans. Neural Networks*, 2007.

[8] C. Williams and M. Seeger, "Using the nystrom method to speed up kernel machines," *NIPS*, 2001.

[9] Kai Zhang, I. W.Tsang, and J. T. Kwok, "Improved Nyström low-rank approximation and error analysis," *ICML*, 2008.