Procedia Computer Science
Volume 51, 2015, Pages 2683–2687
ICCS 2015 International Conference On Computational Science

Scalable Multilevel Support Vector Machines

Talayeh Razzaghi[1] and Ilya Safro[1]

Clemson University, Clemson, SC
(trazzag,isafro)@clemson.edu

Abstract
Solving optimization models (including parameters fitting) for support vector machines on large-scale training data is often an expensive computational task. This paper proposes a multilevel algorithmic framework that scales efficiently to very large data sets. Instead of solving the whole training set in one optimization process, the support vectors are obtained and gradually refined at multiple levels of coarseness of the data. Our multilevel framework substantially improves the computational time without loosing the quality of classifiers. The algorithms are demonstrated for both regular and weighted support vector machines for balanced and imbalanced classification problems. Quality improvement on several imbalanced data sets has been observed.

Keywords: classification, scalable support vector machines, multilevel techniques

1 Introduction

Training nonlinear support vector machines (SVM) is often a time consuming task when the data is big. This problem becomes extremely sensitive when the model selection techniques are applied as both quality, and scalability of SVM depend on the employed optimization solvers. In this paper, we focus on SVMs and weighted SVMs (WSVM) for balanced, and imbalanced data, respectively, that are formulated as the convex quadratic programming (QP). Usually, the complexity required to solve such SVMs is between $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$. We propose a novel method for efficient solution of (W)SVM. In the heart of this method lies a multilevel algorithmic framework (MF) inspired by the multiscale optimization strategies [1]. The main objective of MF is to construct a hierarchy of problems (coarsening), each approximating the original problem but with fewer degrees of freedom. This is achieved by introducing a chain of successive projections of the problem domain into lower-dimensional or smaller-size domains and solving the problem in them using local processing (uncoarsening). The MF combines solutions achieved by the local processing at different levels of coarseness into one global solution. Such frameworks have several key advantages such as a linear complexity, relatively easy parallelization, and adaptivity to hybrid methods with other algorithms. These frameworks are extremely successful in various practical machine learning and data mining tasks such as clustering and dimensionality reduction.

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2015

doi:10.1016/j.procs.2015.05.381

Problem Definition. Let a set of labeled data points be represented by a set $\mathcal{J} = \{(x_i, y_i)\}_{i=1}^{l}$, where $(x_i, y_i) \in \mathbb{R}^{n+1}$, and l and n are the numbers of data points and features, respectively. Each x_i is a data point with n features, and a class label $y_i \in \{-1, 1\}$. An optimal classifier is determined by the parameters w and b through solving the convex problem:

$$\min \quad \frac{1}{2}\,\|w\|^2 + C\sum_{i=1}^{l}\xi_i \quad \text{s.t. } \forall i = 1,\dots,l \quad y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \qquad (1)$$

where ϕ maps training instances x_i into a higher dimensional space, $\phi : \mathbb{R}^n \to \mathbb{R}^m$ $(m \geq n)$. The term slack variables ξ_i $(i \in \{1,\dots,l\})$ in the objective function is used to penalize misclassified points. This approach is also known as *soft margin* SVM. The magnitude of penalization is controlled by the parameter C. The WSVM (an extension of the SVM for imbalanced classes) assigns different weights to each data sample based on its importance, i.e., the objective of (1) is substituted with $\frac{1}{2}\,\|w\|^2 + C^+\sum_{\{i|y_i=+1\}}^{|\mathbf{C}^+|}\xi_i + C^-\sum_{\{j|y_j=-1\}}^{|\mathbf{C}^-|}\xi_j$, where subsets of $\mathcal{J}$ related to the "majority" and "minority" classes are denoted by $\mathbf{C}^-$, and $\mathbf{C}^+$, respectively, i.e., $\mathcal{J} = \mathbf{C}^+ \cup \mathbf{C}^-$. The importance factors C^-, and C^+ are associated with majority and minority classes $\mathbf{C}^-$ and $\mathbf{C}^+$, respectively. In our solvers we employ the Gaussian kernel, and an adapted nested uniform design model selection algorithm [3] for tuning C, C^+, and C^-.

2 Multilevel Support Vector Machines

The proposed MF (see Figure 1) includes three phases: gradual training set coarsening, coarsest support vectors' learning, and gradual support vectors' refinement (uncoarsening). Separate coarsening hierarchies are created for $\mathbf{C}^+$, and $\mathbf{C}^-$ independently. Each next-coarser level contains a subset of points of the corresponding fine level. These subsets are selected using the approximated k-nearest neighbor graphs (AkNN). In contrast to the coarsening used in multilevel dimensionality reduction method [6], we found that selecting an *independent* set only does not lead to the best computational results. Instead, making the coarsening less aggressive makes the framework much more robust to the changes in the parameters. After the coarsest level is solved exactly, we gradually refine the support vectors and the corresponding classifiers.

The Coarsening Phase. The coarsening algorithms are the same for both $\mathbf{C}^+$, and $\mathbf{C}^-$, so we provide only one of them. Given a class of data points $\mathbf{C}$, the coarsening begins with a construction of AkNN $G = (V, E)$, where $V = \mathbf{C}$, and E are the edges of AkNN. The goal is to select a set of points $\hat{V}$ for the next-coarser problem, where $|\hat{V}| \geq Q|V|$ (typically $Q = 0.5$). The second requirement for $\hat{V}$ is that it has to be a dominating set of V. The coarsening for class $\mathbf{C}$ is presented in Algorithm 1. It consists of several iterations of independent set of V selections that are complementary to already chosen sets. We begin with choosing a random independent set (l. 2) using greedy algorithm. It is eliminated from the graph, and the next independent set is added to $\hat{V}$ (l. 5-9). For imbalanced cases, when WSVM is used, we avoid of creating very small coarse problems for $\mathbf{C}^+$. Instead, already very small class is continuously replicated across the rest of the hierarchy if $\mathbf{C}^-$ still requires coarsening. We note that this method of coarsening will reduce the degree of skewness in the data and make the data approximately balanced at the coarsest level. The multilevel framework recursively calls the coarsening process until it creates a hierarchy of r coarse representations $\{\mathcal{J}_i\}_{i=0}^{r}$ of $\mathcal{J}$. At each level of this hierarchy, the corresponding AkNNs' $\{G_i = (V_i, E_i)\}_{i=0}^{r}$ are saved for future use at the uncoarsening phase. The corresponding data and labels at level i is denoted by $(X_i, Y_i) \in \mathbb{R}^{k\times(n+1)}$, where $|X_i| = k$.

Algorithm 1 The Coarsening

Input: $G = (V, E)$ for class $\mathbf{C}$
$\hat{V} \leftarrow$ select maximal independent set in G
$\hat{U} \leftarrow V \setminus \hat{V}$
while $|\hat{V}| < Q \cdot |V|$ **do**
  while $\hat{U} \neq \emptyset$ **do**
    randomly pick $i \in \hat{U}$; $\hat{U} \leftarrow \hat{U} \setminus \{i\}$
    $\hat{U} \leftarrow \hat{U} \setminus \{\text{neighbors of } i \text{ in } \hat{U}\}$
    $\hat{V} \leftarrow \hat{V} \cup \{i\}$
  end while
  $\hat{U} \leftarrow V \setminus \hat{V}$
end while
return $\hat{V}$

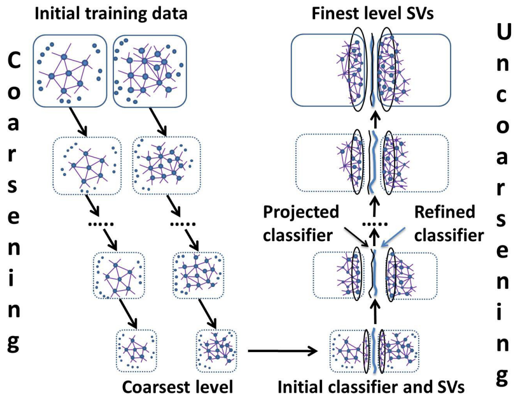

Figure 1: The MF for (W)SVM.

The Coarsest Problem. At the coarsest level r, when $|\mathcal{J}_r| << \mathcal{J}$, we can apply an exact algorithm for training the coarsest classifier. Processing the coarsest level includes an application of the UD [3] model selection to get high-quality classifiers on the difficult data sets.

The Refinement Phase. Given the solution of coarse level $i + 1$ (the set of support vectors S_{i+1}, and parameters C_{i+1}, and γ_{i+1}), the primary goal of the refinement is to update and optimize this solution for the current fine level i. Unlike many other multilevel algorithms, in which the inherited coarse solution contains projected variables only, we initially inherit not only the coarse support vectors but also parameters for model selection. This is because the model selection is an extremely time-consuming component of (W)SVM, and can be prohibitive at fine levels. However, at the coarse levels, when the problem is much smaller than the original, we can apply much heavier methods for model selection without any loss in the total complexity.

Algorithm 2 The Refinement at level i

1: **Input:** $\mathcal{J}_i, S_{i+1}, C_{i+1}, \gamma_{i+1}$
2: **if** i is the coarsest level **then**
3:   Calculate the best (C_i, γ_i) using UD
4:   $S_i \leftarrow$ Apply SVM on X_i
5: **end if**
6: Calculate nearest neighbors N_i for support vectors S_{i+1} from the existing AkNN G_i
7: $data_{train}^{(i)} \leftarrow S_{(i+1)} \cup N_i$
8: **if** $|data_{train}^{(i)}| < Q_{dt}$ **then**
9:   $C^O \leftarrow C_{i+1}$; $\gamma^O \leftarrow \gamma_{i+1}$
10:   Run UD using the center (C^O, γ^O)
11: **else**
12:   $C_i \leftarrow C_{i+1}$; $\gamma_i \leftarrow \gamma_{i+1}$
13: **end if**
14: **if** $|data_{train}^{(i)}| \geq Q_{dt}$ **then**
15:   Cluster $data_{train}^{(i)}$ into K clusters
16:   $\forall k \in K$ find P nearest opposite-class clusters
17:   $S_i \leftarrow$ Apply SVM on pairs of nearest clusters only
18: **else**
19:   $S_i \leftarrow$ Apply SVM directly on $data_{train}^{(i)}$
20: **end if**
21: **Return** S_i, C_i, γ_i

The refinement is presented in Algorithm 2. The coarsest level is solved exactly and reinforced by the model selection (l. 2-5). If i is one of the intermediate levels, we build the set of training data $data_{train}^{(i)}$ by inheriting the coarse support vectors S_{i+1} and adding to them some of their approximated nearest neighbors at level i (l. 6-7) (in our experiments, usually not more

Table 1: Benchmark datasets and computational time (in sec.) of multilevel, and regular SVM.

Dataset	r_{imb}	n_f	$\lvert\mathcal{J}\rvert$	$\lvert\mathbf{C}^+\rvert$	$\lvert\mathbf{C}^-\rvert$	Multilevel Yes ModelSelection		Multilevel No ModelSelection	
						Yes	No	Yes	No
Letter26	0.96	16	20000	734	19266	**45**	112	333	27
Ringnorm	0.50	20	7400	3664	3736	**4**	21	42	12
Twonorm	0.50	20	7400	3703	3697	**4**	21	45	12
Buzz	0.80	77	140707	27775	112932	**2329**	**2400**	70452	20386
Clean (Musk)	0.85	166	6598	1017	5581	**30**	92	167	55
Advertisement	0.86	1558	3279	459	2820	**196**	104	412	41
ISOLET	0.96	617	6238	240	5998	**69**	373	1367	297
cod-rna	0.67	8	59535	19845	39690	**172**	293	1611	208
Nursery	0.67	8	12960	4320	8640	**63**	37	519	42
EEG Eye State	0.55	14	14980	6723	8257	**51**	32	447	33
Hypothyroid	0.94	21	3919	240	3679	**3**	3	5	1

than 5). If the size of $data^{(i)}_{train}$ is still small enough (relatively to the existing computational resources, and the initial size of the data) for applying model selection, and solving SVM on the whole $data^{(i)}_{train}$, then we use coarse parameters C_{i+1}, and γ_{i+1} as initializers for the current level, and retrain (l. 9-10,19). Otherwise, the coarse C_{i+1}, and γ_{i+1} are inherited in C_i, and γ_i (l. 12). Then, being large for direct application of the SVM, $data^{(i)}_{train}$ is clustered into several clusters, and pairs of nearest opposite clusters are retrained, and contribute their solutions to S_i (l. 15-17). *We note that cluster-based retraining can be done in parallel, as different pairs of clusters are independent. Moreover, the total complexity of the algorithm does not suffer from reinforcing the cluster-based retraining with model selection.*

3 Computational Results

Discussion and full results of our work can be found in [5]. The multilevel (W)SVM are evaluated on binary classification benchmarks from UCI repository. Single SVM and WSVM models are solved using LIBSVM-3.18 [2], and the AkNN graphs are costructed using FLANN library [4]. Multilevel frameworks are implemented in MATLAB 2012a, and evaluated on Linux. The results for multilevel (W)SVM (objectives and running times) should only be considered qualitatively and can certainly be further improved by a more advanced implementation. The implementation is available at `http://www.cs.clemson.edu/~isafro/software.html`. Evaluation of the proposed algorithm is done using accuracy (ACC), sensitivity (SN), specificity (SP), and the geometric mean of SN and SP (G-mean). The details of the datasets are described in left part of Table 1. We normalize all data prior to classification in order to get zero mean and unitary standard deviation. We perform a 9- and 5-point run design for the first and second stages of the nested UD.

The performance measures of the multilevel (W)SVM (Table 2, left part of the table) are compared with the regular (one-level) (W)SVM (Table 2, right part of the table). Since several components in the coarsening, and uncoarsening schemes are randomized algorithms, the average numbers over 100 random runs are reported for each data set. We do not report the standard deviations because in all experiments they are negligibly small. Bold fonts emphasize the best G-mean results. Table 2 demonstrates that the quality of multilevel SVM algorithms is very similar to the quality of the single-level SVM. However, we observed that multilevel WSVM improves the single-level WSVM for some datasets.

The main achievement of the proposed multilevel scheme is its computational time (see Table 1) that substantially improves that of the single-level (W)SVM when the model selection techniques must be applied on difficult data sets. We note that for most of the datasets in the benchmark, using model selection was extremely important for obtaining high-quality results. Moreover, the observed improvement is not complete, because (similar to many multilevel and multigrid algorithms) the refinement phase can be easily parallelized at levels where the training by clusters is employed. In addition, the proposed methodology is very successful for large imbalanced classification problems since it can reduce the degree of skewness in the data and make the data approximately balanced at the coarse levels.

Table 2: Performance measures for multilevel and regular SVMs and WSVMs. Each cell contains an average over 100 executions. Column 'Depth' shows the number of levels.

		Multilevel					Not Multilevel			
	Dataset	ACC	SN	SP	G-mean	Depth	ACC	SN	SP	G-mean
SVM	Letter26	0.98	0.99	0.95	0.97	8	1.00	1.00	0.97	0.98
	Ringnorm	0.98	0.98	0.99	**0.98**	6	0.98	0.99	0.98	0.98
	Buzz	0.94	0.96	0.85	**0.90**	14	0.97	0.99	0.81	0.89
	Clean (Musk)	1.00	1.00	0.99	**0.99**	5	1.00	1.00	0.98	0.99
	Advertisement	0.94	0.97	0.79	**0.87**	7	0.92	0.99	0.45	0.67
	ISOLET	0.99	1.00	0.83	**0.92**	11	0.99	1.00	0.85	0.92
	cod-rna	0.95	0.93	0.97	0.95	9	0.96	0.96	0.95	0.96
	Twonorm	0.97	0.98	0.97	0.97	6	0.98	0.98	0.99	0.98
	Nursery	1.00	0.99	0.98	0.99	10	1.00	1.00	1.00	1.00
	EEG Eye State	0.83	0.82	0.88	0.85	6	0.88	0.90	0.86	0.88
	Hypothyroid	0.98	0.98	0.74	**0.85**	4	0.99	1.00	0.71	0.83
WSVM	Letter26	0.99	0.99	0.96	**0.99**	8	1.00	1.00	0.97	0.99
	Ringnorm	0.98	0.97	0.99	**0.98**	6	0.98	0.99	0.98	0.98
	Buzz	0.94	0.96	0.87	**0.91**	14	0.96	0.99	0.81	0.89
	Clean (Musk)	1.00	1.00	0.99	**0.99**	5	1.00	1.00	0.98	0.99
	Advertisement	0.94	0.968	0.80	**0.88**	7	0.92	0.99	0.45	0.67
	ISOLET	0.99	1.00	0.85	**0.92**	11	0.99	1.00	0.85	0.92
	cod-rna	0.94	0.97	0.95	**0.96**	9	0.96	0.96	0.96	0.96
	Twonorm	0.97	0.98	0.97	**0.97**	6	0.98	0.98	0.99	0.98
	Nursery	1.00	0.99	0.98	0.99	10	1.00	1.00	1.00	1.00
	EEG Eye State	0.87	0.89	0.86	**0.88**	6	0.88	0.90	0.86	0.88
	Hypothyroid	0.98	0.98	0.75	**0.86**	4	0.99	1.00	0.75	0.86

References

[1] A. Brandt and D. Ron. Chapter 1 : Multigrid solvers and multilevel optimization strategies. In J. Cong and J. R. Shinnerl, editors, *Multilevel Optimization and VLSICAD*. Kluwer, 2003.

[2] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[3] C.M. Huang, Y.J. Lee, D.K.J. Lin, and S.Y. Huang. Model selection for support vector machines via uniform design. *Computational Statistics & Data Analysis*, 52(1):335–346, 2007.

[4] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09)*, pages 331–340. INSTICC Press, 2009.

[5] T. Razzaghi and I. Safro. Fast multilevel support vector machines. *ArXiv*, abs/1410.3348, 2014.

[6] S. Sakellaridi, Haw ren Fang, and Y. Saad. Graph-based multilevel dimensionality reduction with applications to eigenfaces and latent semantic indexing. In *Machine Learning and Applications, 2008. ICMLA '08. Seventh International Conference on*, pages 194–200, Dec 2008.