

8. Appendix

8.1. Proof of Lemma 1

Proof. When using \bar{K} defined in (3), the matrix Q in (1) becomes \bar{Q} as given below:

$$\bar{Q}_{i,j} = \begin{cases} y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) & \text{if } \pi(\mathbf{x}_i) = \pi(\mathbf{x}_j) \\ 0 & \text{if } \pi(\mathbf{x}_i) \neq \pi(\mathbf{x}_j). \end{cases} \quad (7)$$

Therefore, the quadratic term in (1) can be decomposed into

$$\boldsymbol{\alpha}^T \bar{Q} \boldsymbol{\alpha} = \sum_{c=1}^k \boldsymbol{\alpha}_{(c)}^T Q_{(c,c)} \boldsymbol{\alpha}_{(c)}.$$

The constraints and linear term in (1) are also decomposable, so the subproblems are independent, and concatenation of their optimal solutions, $\bar{\boldsymbol{\alpha}}$, is the optimal solution for (1) when \bar{K} is replaced by \bar{K} . \square

8.2. Proof of Theorem 1

Proof. We use $\bar{f}(\boldsymbol{\alpha})$ to denote the objective function of (1) with kernel \bar{K} . By Lemma 1, $\bar{\boldsymbol{\alpha}}$ is the minimizer of (1) with K replaced by \bar{K} , thus $\bar{f}(\bar{\boldsymbol{\alpha}}) \leq \bar{f}(\boldsymbol{\alpha}^*)$. By the definition of $\bar{f}(\boldsymbol{\alpha}^*)$ we can easily show

$$\bar{f}(\boldsymbol{\alpha}^*) = f(\boldsymbol{\alpha}^*) - \frac{1}{2} \sum_{i,j:\pi(\mathbf{x}_i) \neq \pi(\mathbf{x}_j)} \alpha_i^* \alpha_j^* y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (8)$$

Similarly, we have

$$\bar{f}(\bar{\boldsymbol{\alpha}}) = f(\bar{\boldsymbol{\alpha}}) - \frac{1}{2} \sum_{i,j:\pi(\mathbf{x}_i) \neq \pi(\mathbf{x}_j)} \bar{\alpha}_i \bar{\alpha}_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j). \quad (9)$$

Combining with $\bar{f}(\bar{\boldsymbol{\alpha}}) \leq \bar{f}(\boldsymbol{\alpha}^*)$ we have

$$\begin{aligned} f(\bar{\boldsymbol{\alpha}}) &\leq \bar{f}(\boldsymbol{\alpha}^*) + \frac{1}{2} \sum_{i,j:\pi(\mathbf{x}_i) \neq \pi(\mathbf{x}_j)} \bar{\alpha}_i \bar{\alpha}_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \\ &= f(\boldsymbol{\alpha}^*) + \frac{1}{2} \sum_{i,j:\pi(\mathbf{x}_i) \neq \pi(\mathbf{x}_j)} (\bar{\alpha}_i \bar{\alpha}_j - \alpha_i^* \alpha_j^*) y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\leq f(\boldsymbol{\alpha}^*) + \frac{1}{2} C^2 D(\pi), \text{ since } 0 \leq \bar{\alpha}_i, \alpha_i^* \leq C \text{ for all } i. \end{aligned} \quad (10)$$

Also, since $\boldsymbol{\alpha}^*$ is the optimal solution of (1) and $\bar{\boldsymbol{\alpha}}$ is a feasible solution, $f(\boldsymbol{\alpha}^*) < f(\bar{\boldsymbol{\alpha}})$, thus proving the first part of the theorem.

Let σ_n be the smallest singular value of the positive definite kernel matrix K . Since $Q = \text{diag}(\mathbf{y})K\text{diag}(\mathbf{y})$ and $y_i \in \{1, -1\}$ for all i , Q and K have identical singular values. Suppose we write $\bar{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^* + \Delta\boldsymbol{\alpha}$,

$$f(\bar{\boldsymbol{\alpha}}) = f(\boldsymbol{\alpha}^*) + (\boldsymbol{\alpha}^*)^T Q \Delta\boldsymbol{\alpha} + \frac{1}{2} (\Delta\boldsymbol{\alpha})^T Q \Delta\boldsymbol{\alpha} - e^T \Delta\boldsymbol{\alpha}. \quad (11)$$

The optimality condition for (1) is

$$\nabla_i f(\boldsymbol{\alpha}^*) \begin{cases} = 0 & \text{if } 0 < \alpha_i^* < C, \\ \geq 0 & \text{if } \alpha_i^* = 0, \\ \leq 0 & \text{if } \alpha_i^* = C, \end{cases} \quad (12)$$

where $\nabla f(\boldsymbol{\alpha}^*) = Q\boldsymbol{\alpha}^* - e$. Since $\bar{\boldsymbol{\alpha}}$ is a feasible solution, it is easy to see that $(\Delta\boldsymbol{\alpha})_i \geq 0$ if $\alpha_i^* = 0$, and $(\Delta\boldsymbol{\alpha})_i \leq 0$ if $\alpha_i^* = C$. Thus,

$$(\Delta\boldsymbol{\alpha})^T (Q\boldsymbol{\alpha}^* - e) = \sum_{i=1}^n (\Delta\boldsymbol{\alpha})_i ((Q\boldsymbol{\alpha}^*)_i - 1) \geq 0.$$

Combining with (11) we have $f(\bar{\boldsymbol{\alpha}}) \geq f(\boldsymbol{\alpha}^*) + \frac{1}{2} \Delta\boldsymbol{\alpha}^T Q \Delta\boldsymbol{\alpha} \geq f(\boldsymbol{\alpha}^*) + \frac{1}{2} \sigma_n \|\Delta\boldsymbol{\alpha}\|_2^2$. Since we already know that $f(\bar{\boldsymbol{\alpha}}) \leq f(\boldsymbol{\alpha}^*) + \frac{1}{2} C^2 D(\pi)$, this implies $\|\boldsymbol{\alpha}^* - \bar{\boldsymbol{\alpha}}\|_2^2 \leq C^2 D(\pi) / \sigma_n$. \square

8.3. Proof of Theorem 2

Proof. Let $\Delta Q = Q - \bar{Q}$ and $\Delta\boldsymbol{\alpha} = \boldsymbol{\alpha}^* - \bar{\boldsymbol{\alpha}}$. From the optimality condition for (1) (see (12)), we know that $\alpha_i^* = 0$ if $(Q\boldsymbol{\alpha}^*)_i > 1$. Since $Q\boldsymbol{\alpha}^* = (\bar{Q} + \Delta Q)(\bar{\boldsymbol{\alpha}} + \Delta\boldsymbol{\alpha})$, we see that

$$\begin{aligned} (Q\boldsymbol{\alpha}^*)_i &= (\bar{Q}\bar{\boldsymbol{\alpha}})_i + (\Delta Q\bar{\boldsymbol{\alpha}})_i + (Q\Delta\boldsymbol{\alpha})_i \\ &= (\bar{Q}\bar{\boldsymbol{\alpha}})_i + \sum_{j:\pi(\mathbf{x}_i) \neq \pi(\mathbf{x}_j)} y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \bar{\alpha}_j \\ &\quad + \sum_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) (\Delta\boldsymbol{\alpha})_j \\ &\geq (\bar{Q}\bar{\boldsymbol{\alpha}})_i - CD(\pi) - K_{max} \|\Delta\boldsymbol{\alpha}\|_1 \\ &\geq (\bar{Q}\bar{\boldsymbol{\alpha}})_i - CD(\pi) \\ &\quad - \sqrt{n} K_{max} C \sqrt{D(\pi)} / \sqrt{\sigma_n} \text{ (by Theorem 1)} \\ &= (\bar{Q}\bar{\boldsymbol{\alpha}})_i - CD(\pi) \left(1 + \frac{\sqrt{n} K_{max}}{\sqrt{\sigma_n D(\pi)}} \right). \end{aligned}$$

The condition stated in the theorem implies $(\bar{Q}\bar{\boldsymbol{\alpha}})_i > 1 + CD(\pi) \left(1 + \frac{\sqrt{n} K_{max}}{\sqrt{\sigma_n D(\pi)}} \right)$, which implies $(Q\boldsymbol{\alpha}^*)_i - 1 > 0$, so from the optimality condition (12), $\alpha_i^* = 0$. \square

8.4. Proof of Theorem 3

Proof. Similar to the proof in Theorem 1, we use $\bar{f}(\boldsymbol{\alpha})$ to denote the objective function of (1) with kernel \bar{K} . Combine (10) with the fact that $\alpha_i^* = 0 \forall i \notin S^*$ and $\bar{\alpha}_i = 0 \forall i \notin \bar{S}$, we have

$$\begin{aligned} \bar{f}(\boldsymbol{\alpha}^*) &\leq f(\boldsymbol{\alpha}^*) - \frac{1}{2} \sum_{i,j:\pi(\mathbf{x}_i) \neq \pi(\mathbf{x}_j) \text{ and } i,j \in S^*} (\bar{\alpha}_i \bar{\alpha}_j - \alpha_i^* \alpha_j^*) y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\leq f(\boldsymbol{\alpha}^*) + \frac{1}{2} C^2 D(\{\mathbf{x}_i\}_{i \in S^* \cup \bar{S}}, \pi). \end{aligned}$$

The second part of the proof is exactly the same as the second part of Theorem 1. \square

8.5. Clustering time vs Training time

Our DC-SVM algorithm is composed of two important parts: clustering and SVM training. In Table 5 we list the time taken by each part; we can see that the clustering time is almost constant at each level, while the rest of the training time keeps increasing.

Table 5: Run time (in seconds) for DC-SVM on different levels (covtype dataset). We can see the clustering time is only a small portion compared with the total training time.

Level	4	3	2	1	0
Clustering	43.2s	42.5s	40.8s	38.1s	36.5s
Training	159.4s	439.7s	1422.8s	3135.5s	7614.0s

8.6. Comparison with Bagging Approach

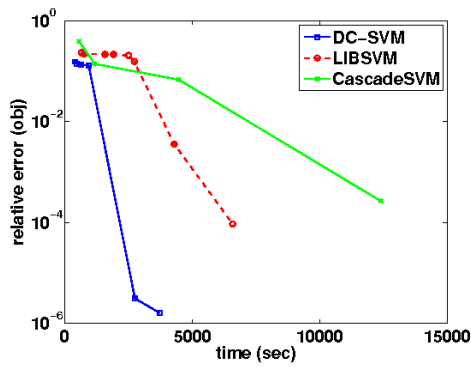
Bootstrap aggregating (bagging) is a machine learning approach designed to improve the stability of machine learning algorithms. Given a training set with n samples, bagging generates k training sets, each by sampling \bar{n} data points uniformly from the whole dataset. Considering the case that $\bar{n} = n/k$, then the bagging algorithms is similar to our DCSVM (early) approach, but with the following two differences:

- Data partition: bagging uses random sampling while DCSVM (early) uses clustering.
- Prediction: bagging uses voting for classification task, while DCSVM (early) using the nearest model for prediction.

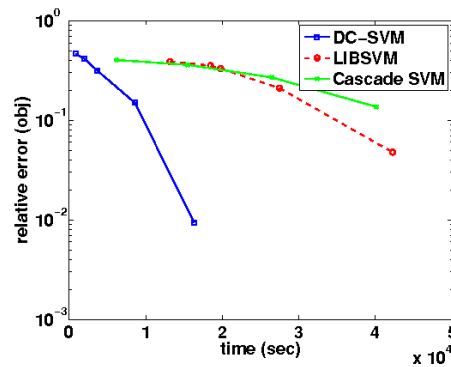
Under the same k , both DCSVM (early) and bagging trains the k subsets independently, so the training times are identical for both algorithms. We compare the classification performance under various values of k in Table 6 on `ijcnn1`, `covtype`, and `webspam` datasets. The results show that DCSVM (early) is significantly better than bagging in terms of prediction accuracy.

Table 6: Prediction accuracy of DC-SVM (early) and bagging under various values of k . We can see that DCSVM (early) is significantly better than bagging.

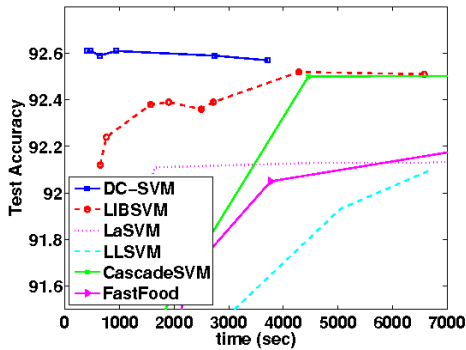
k	ijcnn1		covtype		webspam	
	DCSVM (early)	Bagging	DCSVM (early)	Bagging	DCSVM (early)	Bagging
256	98.16%	91.81%	96.12%	83.41%	99.04%	95.20%
64	98.35%	95.44%	96.15%	88.54%	99.23%	97.13%
16	98.46%	98.24%	96.16%	91.81%	99.29%	98.28%



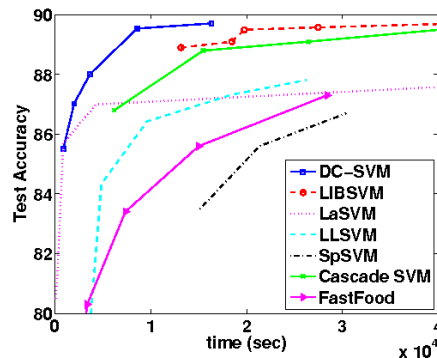
(a) kddcup99 objective function



(b) cifar objective function



(c) kddcup99 testing accuracy



(d) cifar testing accuracy

Figure 5: Additional comparison of algorithms using RBF kernel on the kddcup99 and cifar datasets.

Table 7: Comparison of DC-SVM, DC-SVM (early), and LIBSVM on ijcnn1 with various parameters C, γ . DC-SVM (early) is always 10 times faster than LIBSVM achieves similar testing accuracy. DC-SVM is faster than LIBSVM for almost every setting.

dataset	C	γ	DC-SVM (early)		DC-SVM		LIBSVM		LaSVM	
			acc(%)	time(s)	acc(%)	time(s)	acc(%)	time(s)	acc(%)	time(s)
ijcnn1	2^{-10}	2^{-10}	90.5	12.8	90.5	120.1	90.5	130.0	90.5	492
ijcnn1	2^{-10}	2^{-6}	90.5	12.8	90.5	203.1	90.5	492.5	90.5	526
ijcnn1	2^{-10}	2^1	90.5	50.4	90.5	524.2	90.5	1121.3	90.5	610
ijcnn1	2^{-10}	2^6	93.7	44.0	93.7	400.2	93.7	1706.5	92.4	1139
ijcnn1	2^{-10}	2^{10}	97.1	39.1	97.1	451.3	97.1	1214.7	95.7	1711
ijcnn1	2^{-6}	2^{-10}	90.5	7.2	90.5	84.7	90.5	252.7	90.5	531
ijcnn1	2^{-6}	2^{-6}	90.5	7.6	90.5	161.2	90.5	401.0	90.5	519
ijcnn1	2^{-6}	2^1	90.7	10.8	90.8	183.6	90.8	553.2	90.5	577
ijcnn1	2^{-6}	2^6	93.9	49.2	93.9	416.1	93.9	1645.3	91.3	1213
ijcnn1	2^{-6}	2^{10}	97.1	40.6	97.1	477.3	97.1	1100.7	95.5	1744
ijcnn1	2^1	2^{-10}	90.5	14.0	90.5	305.6	90.5	424.9	90.5	511
ijcnn1	2^1	2^{-6}	91.8	12.6	92.0	254.6	92.0	367.1	90.8	489
ijcnn1	2^1	2^1	98.8	7.0	98.8	43.5	98.8	111.6	95.4	227
ijcnn1	2^1	2^6	98.3	34.6	98.3	584.5	98.3	1776.5	97.8	1085
ijcnn1	2^1	2^{10}	97.2	94.0	97.2	523.1	97.2	1955.0	96.1	1691
ijcnn1	2^6	2^{-10}	92.5	27.8	91.9	276.3	91.9	331.8	90.5	442
ijcnn1	2^6	2^{-6}	94.8	19.9	95.6	313.7	95.6	219.5	92.3	435
ijcnn1	2^6	2^1	98.3	6.4	98.3	75.3	98.3	59.8	97.5	222
ijcnn1	2^6	2^6	98.1	48.3	98.1	384.5	98.1	987.7	97.1	1144
ijcnn1	2^6	2^{10}	97.2	51.9	97.2	530.7	97.2	1340.9	95.4	1022
ijcnn1	2^{10}	2^{-10}	94.4	146.5	92.5	606.1	92.5	1586.6	91.7	401
ijcnn1	2^{10}	2^{-6}	97.3	124.3	97.6	553.6	97.6	1152.2	96.5	1075
ijcnn1	2^{10}	2^1	97.5	10.6	97.5	50.8	97.5	139.3	97.1	605
ijcnn1	2^{10}	2^6	98.2	42.5	98.2	338.3	98.2	1629.3	97.1	890
ijcnn1	2^{10}	2^{10}	97.2	66.4	97.2	309.6	97.2	2398.3	95.4	909

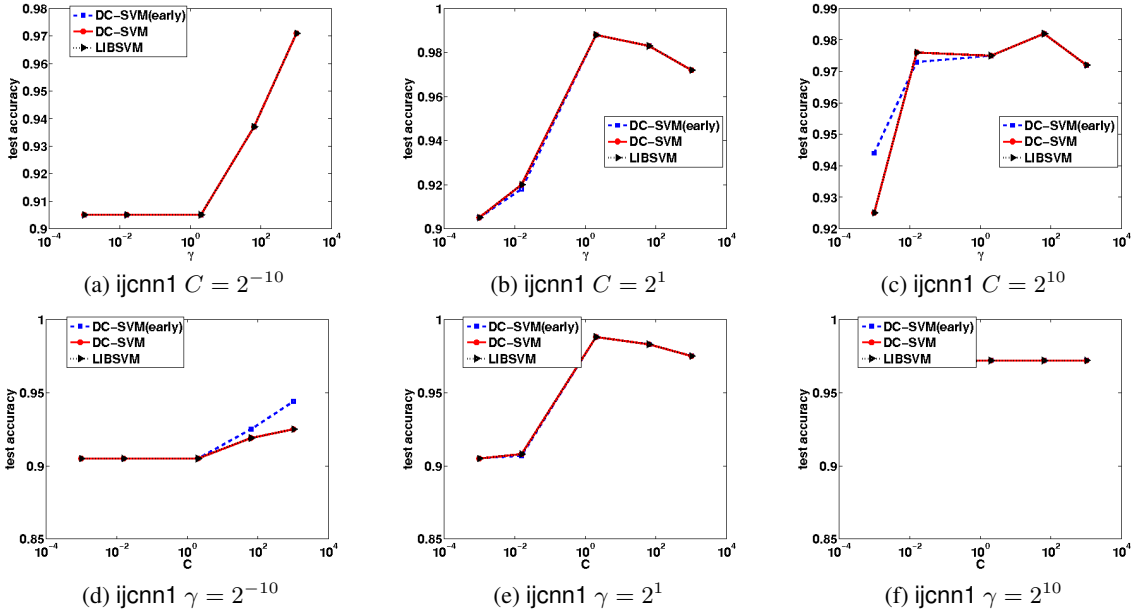


Figure 6: Robustness to the parameters C, γ on ijcnn1 dataset.

Table 8: Comparison of DC-SVM, DC-SVM (early) and LIBSVM on webspam with various parameters C, γ . DC-SVM (early) is always more than 30 times faster than LIBSVM and has comparable or better test accuracy; DC-SVM is faster than LIBSVM under all settings.

dataset	C	γ	DC-SVM (early)		DC-SVM		LIBSVM	
			acc(%)	time(s)	acc(%)	time(s)	acc(%)	time(s)
webspam	2^{-10}	2^{-10}	86	806	61	26324	61	45984
webspam	2^{-10}	2^{-6}	83	935	61	22569	61	53569
webspam	2^{-10}	2^1	87.1	886	91.1	10835	91.1	34226
webspam	2^{-10}	2^6	93.7	1060	92.6	6496	92.6	34558
webspam	2^{-10}	2^{10}	98.3	1898	98.5	7410	98.5	55574
webspam	2^{-6}	2^{-10}	83	793	68	24542	68	44153
webspam	2^{-6}	2^{-6}	84	762	69	33498	69	63891
webspam	2^{-6}	2^1	93.3	599	93.5	15098	93.1	34226
webspam	2^{-6}	2^6	96.4	704	96.4	7048	96.4	48571
webspam	2^{-6}	2^{10}	98.3	1277	98.6	6140	98.6	45122
webspam	2^1	2^{-10}	87	688	78	18741	78	48512
webspam	2^1	2^{-6}	93	645	81	10481	81	30106
webspam	2^1	2^1	98.4	420	99.0	9157	99.0	35151
webspam	2^1	2^6	98.9	466	98.9	5104	98.9	28415
webspam	2^1	2^{10}	98.3	853	98.7	4490	98.7	28891
webspam	2^6	2^{-10}	93	759	80	24849	80	64121
webspam	2^6	2^{-6}	97	602	83	21898	83	55414
webspam	2^6	2^1	98.8	406	99.1	8051	99.1	40510
webspam	2^6	2^6	99.0	465	98.9	6140	98.9	35510
webspam	2^6	2^{10}	98.3	917	98.7	4510	98.7	34121
webspam	2^{10}	2^{-10}	97	1350	82	31387	82	81592
webspam	2^{10}	2^{-6}	98	1127	86	34432	86	82581
webspam	2^{10}	2^1	98.8	463	98.8	10433	98.8	58512
webspam	2^{10}	2^6	99.0	455	99.0	15037	99.0	75121
webspam	2^{10}	2^{10}	98.3	831	98.7	7150	98.7	59126

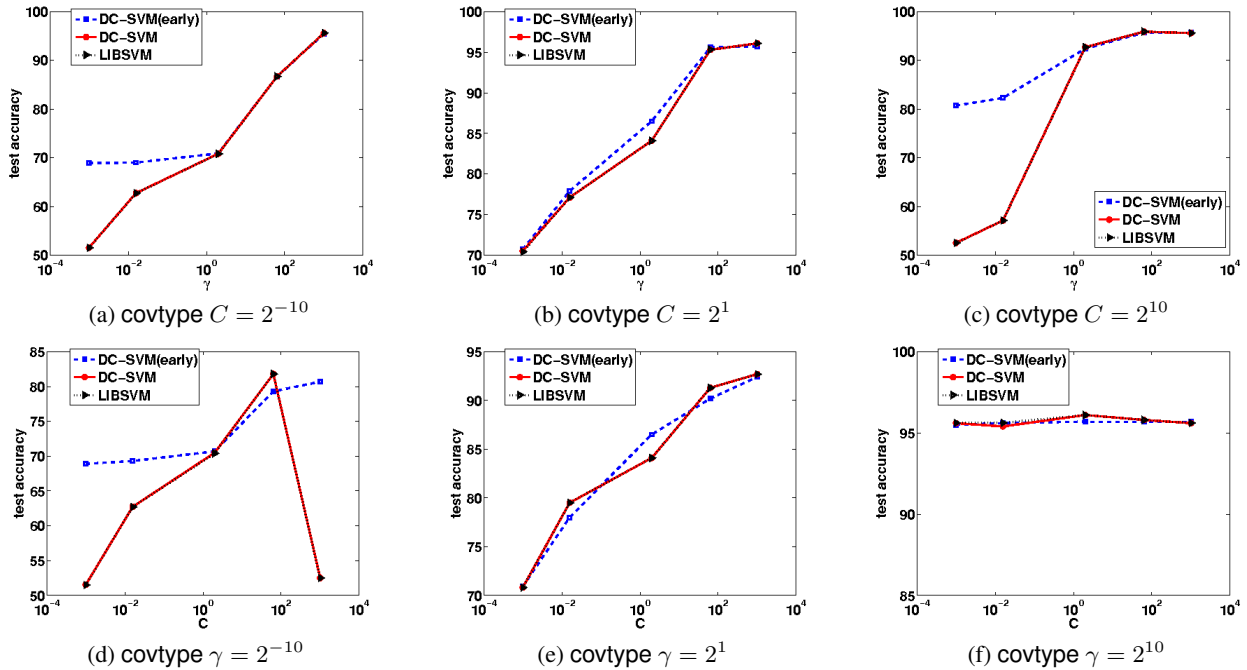


Figure 7: Robustness to the parameters C, γ on covtype dataset.

Table 9: Comparison of DC-SVM, DC-SVM (early) and LIBSVM on covtype with various parameters C, γ . DC-SVM (early) is always more than 50 times faster than LIBSVM with similar test accuracy; DC-SVM is faster than LIBSVM under all settings.

dataset	C	γ	DC-SVM (early)		DC-SVM		LIBSVM	
			acc(%)	time(s)	acc(%)	time(s)	acc(%)	time(s)
covtype	2^{-10}	2^{-10}	68.9	736	51.5	24791	51.5	48858
covtype	2^{-10}	2^{-6}	69.0	507	62.7	17189	62.7	62668
covtype	2^{-10}	2^1	70.9	624	70.8	12997	70.8	88160
covtype	2^{-10}	2^6	86.7	1351	86.7	13985	86.7	85111
covtype	2^{-10}	2^{10}	95.5	1173	95.6	9480	95.6	54282
covtype	2^{-6}	2^{-10}	69.3	373	62.7	10387	62.7	90774
covtype	2^{-6}	2^{-6}	70.0	625	68.6	14398	68.6	76508
covtype	2^{-6}	2^1	78.0	346	79.5	5312	79.5	77591
covtype	2^{-6}	2^6	87.9	895	87.9	8886	87.9	120512
covtype	2^{-6}	2^{10}	95.6	1238	95.4	7581	95.6	123396
covtype	2^1	2^{-10}	70.7	433	70.4	25120	70.4	88725
covtype	2^1	2^{-6}	77.9	1000	77.1	18452	77.1	69101
covtype	2^1	2^1	86.5	421	84.1	11411	84.1	50890
covtype	2^1	2^6	95.6	299	95.3	8714	95.3	117123
covtype	2^1	2^{10}	95.7	882	96.1	5349		>300000
covtype	2^6	2^{-10}	79.3	1360	81.8	34181	81.8	105855
covtype	2^6	2^{-6}	81.3	2314	84.3	24191	84.3	108552
covtype	2^6	2^1	90.2	957	91.3	14099	91.3	75596
covtype	2^6	2^6	96.3	356	96.2	9510	96.2	92951
covtype	2^6	2^{10}	95.7	961	95.8	7483	95.8	288567
covtype	2^{10}	2^{-10}	80.7	5979	52.5	50149	52.5	235183
covtype	2^{10}	2^{-6}	82.3	8306	57.1	43488		> 300000
covtype	2^{10}	2^1	92.4	4553	92.7	19481	92.7	254130
covtype	2^{10}	2^6	95.7	368	95.9	12615	95.9	93231
covtype	2^{10}	2^{10}	95.7	1094	95.6	10432	95.6	169918

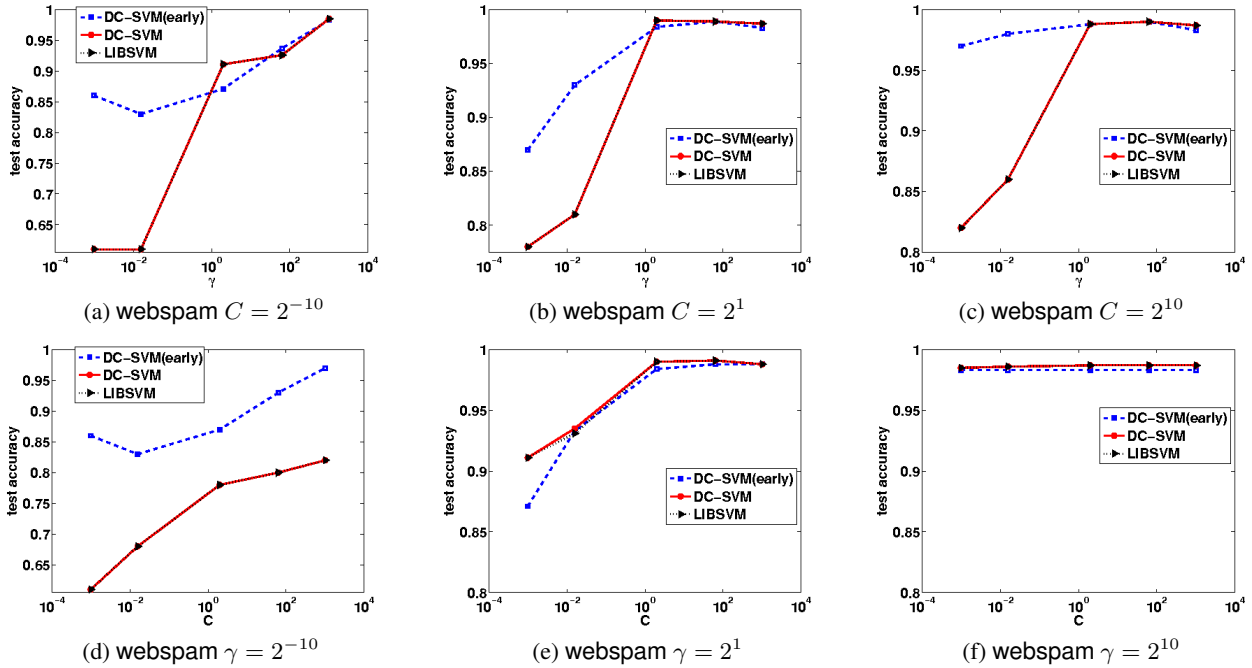


Figure 8: Robustness to the parameters C, γ on webspam dataset.

Table 10: Comparison of DC-SVM, DC-SVM (early) and LIBSVM on census with various parameters C, γ . DC-SVM (early) is always more than 50 times faster than LIBSVM with similar test accuracy; DC-SVM is faster than LIBSVM under all settings.

dataset	C	γ	DC-SVM (early)		DC-SVM		LIBSVM	
			acc(%)	time(s)	acc(%)	time(s)	acc(%)	time(s)
census	2^{-10}	2^{-10}	93.80	161	93.80	2153	93.80	3061
census	2^{-10}	2^{-6}	93.80	166	93.80	3316	93.80	5357
census	2^{-10}	2^1	93.61	202	93.68	4215	93.66	11947
census	2^{-10}	2^6	91.96	228	92.08	5104	92.08	12693
census	2^{-10}	2^{10}	62.00	195	56.32	4951	56.31	13604
census	2^{-6}	2^{-10}	93.80	145	93.80	3912	93.80	6693
census	2^{-6}	2^{-6}	93.80	149	93.80	3951	93.80	6568
census	2^{-6}	2^1	93.63	217	93.66	4145	93.66	11945
census	2^{-6}	2^6	91.97	230	92.10	4080	92.10	9404
census	2^{-6}	2^{10}	62.58	189	56.32	3069	56.31	9078
census	2^1	2^{-10}	93.80	148	93.95	2057	93.95	1908
census	2^1	2^{-6}	94.55	139	94.82	2018	94.82	1998
census	2^1	2^1	93.27	179	93.36	4031	93.36	37023
census	2^1	2^6	91.96	220	92.06	6148	92.06	33058
census	2^1	2^{10}	62.78	184	56.31	6541	56.31	35031
census	2^6	2^{-10}	94.66	193	94.66	3712	94.69	3712
census	2^6	2^{-6}	94.76	164	95.21	2015	95.21	3725
census	2^6	2^1	93.10	229	93.15	6814	93.15	32993
census	2^6	2^6	91.77	243	91.88	9158	91.88	34035
census	2^6	2^{10}	62.18	210	56.25	9514	56.25	36910
census	2^{10}	2^{-10}	94.83	538	94.83	2751	94.85	8729
census	2^{10}	2^{-6}	93.89	315	92.94	3548	92.94	12735
census	2^{10}	2^1	92.89	342	92.92	9105	92.93	52441
census	2^{10}	2^6	91.64	244	91.81	7519	91.81	34350
census	2^{10}	2^{10}	61.14	206	56.25	5917	56.23	34906

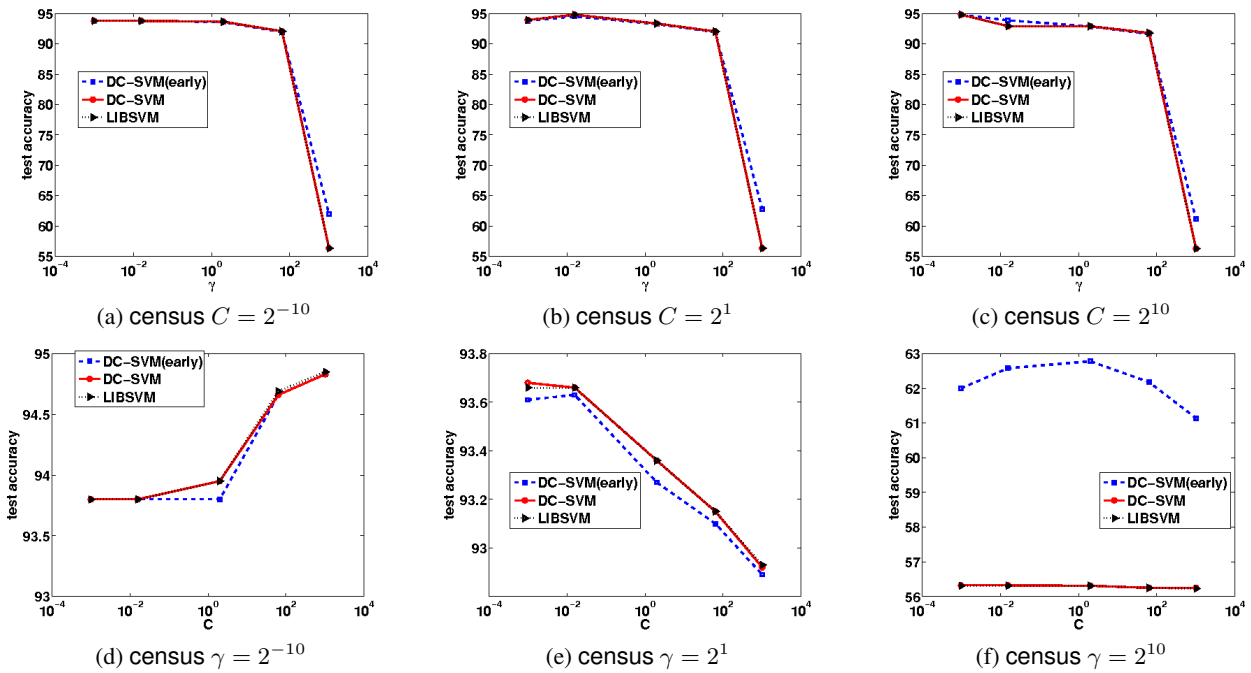


Figure 9: Robustness to the parameters C, γ on census dataset.