Supplementary Material

A SVM Sub-steps for Consensus Optimization

A common formulation of the support vector machine (SVM) solves

$$\text{minimize} \quad \frac{1}{2}\|x\|^2 + Ch(Dx) \tag{16}$$

where C is a regularization parameter, and h is a simple "hinge loss" function given by $h(z) = \sum_{k=1}^{M} \max\{1 - l_k z_k, 0\}$. The proximal mapping of h has the form $\text{prox}_h(z,\delta)_k = z_k + l_k \max\{\min\{1 - l_k z_k, \delta\}, 0\}$. Using this proximal operator, the solution to the y update in (10) is simply $y^{k+1} = \text{prox}_h\left(Dx^{k+1} + \lambda^k, \frac{C}{\tau}\right)$. Note that this algorithm is much simpler than the consensus implementation of SVM, which requires each node to solve the sub-problem

$$\text{minimize} \quad Ch(Dx) + \frac{\tau}{2}\|x - y\|^2. \tag{17}$$

Despite the similarity of this problem to the original SVM (16), this problem form is *not* supported by available SVM solvers such as LIBSVM [23] and others. However, techniques for the classical SVM problem can be easily adapted to solve (17).

A common numerical approach to solving (16) is the attack its dual, which is

$$\underset{\alpha_i \in [0,C]}{\text{minimize}} \quad \frac{1}{2}\|A^T L\alpha\|^2 - \alpha^T \mathbf{1} = \sum_{i,j} \alpha_i \alpha_j l_i l_j A_i A_j^T - \sum_i \alpha_i. \tag{18}$$

Once (18) is solved to obtain $\alpha^\star$, the solution to (13) is simply given by $w^\star = L^T\alpha$. The dual formulation (18) is advantageous because the constraints on α act separately on each coordinate. The dual is therefore solved efficiently by coordinate descent, which is the approach used by the popular solver LIBSVM [23]. This method is particularly powerful when the number of support vectors in the solution is small, in which case most of the entries in α assume the value 0 or C.

In the context of consensus ADMM, we must solve

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + Ch(Aw, l) + \frac{\tau}{2}\|w - z\|^2. \tag{19}$$

Following the classical SVM literature, we dualize this problem to obtain

$$\underset{\alpha_i \in [0,C]}{\text{minimize}} \quad \frac{1}{2}\|A^T L\alpha\|^2 - \alpha^T\left((1+\tau)\mathbf{1} - \tau L z\right). \tag{20}$$

We then solve (20) for $\alpha^\star$, and recover the solution via

$$w^\star = \frac{A^T L\alpha + \tau z}{1+\tau}.$$

We solve (20) using a dual coordinate descent method inspired by [23]. The implementation has $O(M)$ complexity per iteration. Also following [23] we optimize the convergence by updating coordinates with the largest residual (derivative) on each pass.

Because our solver does not need to handle a "bias" variable (in consensus optimization, only the central server treats the bias variables differently from other unknowns), and by using a warm start to accelerate solve time across iterations, our coordinate descent method significantly outperforms even LIBSVM for each sub-problem. On a desktop computer with a Core i5 processor, LIBSVM solves the synthetic data test problem with $m = 100$ datapoints and $n = 200$ features in 3.4 seconds (excluding "setup" time), as opposed to our custom solver which solves each SVM sub-problem for the consensus SVM with the same dimensions (on a single processor) in 0.17 seconds (averaged over all iterations). When $m = 10000$ and $n = 20$, LIBSVM requires over 20 seconds, while the average solve time for the custom solver embedded in the consensus method is only 2.3 seconds.

B Tables of Results

In the following tables, we use these labels:

- N: Number of data points per core
- F: Number of features per data point
- Cores: Number of compute cores used in computation
- Space: Total size of data corpus in GB (truncated at GB)
- TWalltime: Walltime for transpose method (truncated at seconds)
- TCompute: Total computation time for transpose method (truncated at seconds)
- CWalltime: Walltime for consensus method (truncated at seconds)
- CCompute: Total computation time for consensus method (truncated at seconds)

Logistic regression with homogeneous data

N	F	Cores	Space(GB)	TWalltime	TCompute	CWalltime	CCompute
50000	2000	800	596	0:00:53	6:19:14	0:01:36	17:25:18
50000	2000	1600	1192	0:00:58	12:40:24	0:01:51	1 day 10:51:33
50000	2000	2400	1788	0:01:00	19:05:13	0:01:52	2 days 4:21:25
50000	2000	3200	2384	0:01:00	1 day 1:30:18	0:01:41	2 days 21:46:28
50000	2000	4000	2980	0:00:58	1 day 7:58:24	0:01:39	3 days 15:17:51
50000	2000	4800	3576	0:00:58	1 day 14:27:31	0:02:31	4 days 8:49:58
50000	2000	5600	4172	0:01:00	1 day 21:10:38	0:02:13	5 days 2:16:56
50000	2000	6400	4768	0:01:03	2 days 3:46:42	0:02:08	5 days 19:39:40
50000	2000	7200	5364	0:01:21	2 days 10:36:36	0:01:47	6 days 13:12:59
100000	1000	2000	1490	0:02:09	11:50:56	0:01:58	2 days 0:28:08
100000	1000	4000	2980	0:01:32	1 day 0:05:30	0:04:14	4 days 0:58:47
100000	1000	6000	4470	0:01:40	1 day 12:20:57	0:02:00	6 days 1:36:20
100000	1000	8000	5960	0:00:42	2 days 0:42:49	0:03:33	8 days 1:59:14
100000	1000	10000	7450	0:01:01	3 days 5:30:41	0:02:43	10 days 2:30:10
100000	1000	12000	8940	0:01:16	4 days 0:50:36	0:02:54	12 days 2:59:08
100000	1000	14000	10430	0:01:33	4 days 16:42:20	0:05:00	14 days 3:36:58
100000	1000	16000	11920	0:01:18	5 days 3:40:44	0:03:19	16 days 4:11:34
100000	1000	18000	13411	0:01:07	5 days 6:45:44	0:05:29	18 days 4:56:02
100000	1000	20000	14901	0:01:17	6 days 21:44:52	0:03:14	20 days 5:36:16
5000	2000	4800	357	0:00:33	4:04:11	0:00:26	21:01:22
10000	2000	4800	715	0:00:26	7:51:06	0:01:22	1 day 21:24:47
15000	2000	4800	1072	0:00:38	11:23:22	0:01:37	2 days 19:42:30
20000	2000	4800	1430	0:00:42	15:15:01	0:01:30	3 days 19:27:24
25000	2000	4800	1788	0:00:42	18:59:04	0:01:48	4 days 17:24:59
30000	2000	4800	2145	0:00:47	22:53:25	0:02:04	5 days 16:30:28
35000	2000	4800	2503	0:00:57	1 day 2:43:48	0:02:46	6 days 15:10:40
40000	2000	4800	2861	0:00:54	1 day 6:22:51	0:02:42	7 days 14:58:02
45000	2000	4800	3218	0:00:57	1 day 10:05:17	0:03:02	8 days 15:11:42
50000	2000	4800	3576	0:01:02	1 day 14:28:30	0:03:24	9 days 15:51:21
20000	500	4800	357	0:00:05	2:18:21	0:00:35	20:51:20
20000	1000	4800	715	0:00:12	5:33:31	0:01:40	1 day 18:43:21
20000	1500	4800	1072	0:00:25	9:44:07	0:01:08	2 days 20:08:20
20000	2000	4800	1430	0:00:31	15:10:01	0:01:29	3 days 19:28:56
20000	2500	4800	1788	0:01:23	1 day 12:24:25	0:03:30	4 days 20:53:53
20000	3000	4800	2145	0:01:50	1 day 20:29:59	0:03:44	5 days 19:45:31
20000	3500	4800	2503	0:02:27	2 days 5:40:09	0:03:56	6 days 19:44:54
20000	4000	4800	2861	0:03:03	2 days 16:50:51	0:03:46	7 days 18:17:21
20000	4500	4800	3218	0:04:00	3 days 3:35:02	0:04:28	8 days 19:49:26
20000	5000	4800	3576	0:04:52	3 days 16:50:21	0:04:44	9 days 23:56:16

Logistic regression with heterogeneous data

N	F	Cores	Space(GB)	TWalltime	TCompute	CWalltime	CCompute
50000	2000	800	596	0:00:56	6:14:57	0:09:25	3 days 19:56:38
50000	2000	1600	1192	0:01:01	12:28:12	0:09:35	7 days 19:00:17
50000	2000	2400	1788	0:00:58	18:43:11	0:09:35	11 days 13:26:10
50000	2000	3200	2384	0:00:58	1 day 1:09:09	0:09:39	15 days 10:33:19
50000	2000	4000	2980	0:01:23	1 day 7:34:22	0:09:49	19 days 6:45:31
50000	2000	4800	3576	0:01:11	1 day 13:51:15	0:34:30	77 days 5:23:50
50000	2000	5600	4172	0:01:29	1 day 20:20:50	0:34:38	90 day 19:10:12
50000	2000	6400	4768	0:01:01	2 days 2:55:20	0:35:31	103 days 19:09:22
50000	2000	7200	5364	0:01:14	2 days 9:38:02	0:10:26	34 days 20:11:28
100000	1000	2000	1490	0:01:31	11:15:47	0:26:49	23 days 21:30:59
100000	1000	4000	2980	0:01:03	22:44:45	0:25:23	48 days 17:23:23
100000	1000	6000	4470	0:00:42	1 day 10:38:14	0:24:38	73 days 15:10:07
100000	1000	8000	5960	0:00:43	1 day 22:25:35	0:25:08	98 days 12:53:22
100000	1000	10000	7450	0:00:56	2 days 10:13:27	0:25:39	123 days 0:26:26
100000	1000	12000	8940	0:01:24	2 days 22:10:47	0:25:00	146 days 22:00:35
100000	1000	14000	10430	0:01:16	4 days 11:33:53	0:26:27	171 days 8:40:10
100000	1000	16000	11920	0:00:56	3 days 22:59:09	0:25:18	195 days 19:54:41
100000	1000	18000	13411	0:01:26	4 days 11:34:10	0:26:03	218 days 19:17:19
100000	1000	20000	14901	0:01:59	4 days 23:59:15	0:26:27	243 days 4:55:47

Lasso with heterogeneous data

N	F	Cores	Space(GB)	TWalltime	TCompute	CWalltime	CCompute
50000	200	800	59	0:00:12	0:01:45	0:00:37	0:04:55
50000	200	1600	119	0:00:02	0:03:31	0:00:47	0:10:56
50000	200	2400	178	0:00:02	0:05:14	0:01:14	0:17:50
50000	200	3200	238	0:00:00	0:07:00	0:01:22	0:25:24
50000	200	4000	298	0:00:04	0:09:00	0:01:36	0:33:49
50000	200	4800	357	0:00:11	0:10:25	0:01:57	0:43:29
50000	200	5600	417	0:00:10	0:12:09	0:02:07	0:55:47
50000	200	6400	476	0:00:07	0:13:48	0:02:19	1:04:51
50000	200	7200	536	0:00:09	0:15:31	0:02:39	1:19:22
50000	1000	800	298	0:00:04	0:33:28	0:05:20	2:58:02
50000	1000	1600	596	0:00:18	1:06:33	0:06:23	6:00:37
50000	1000	2400	894	0:00:25	1:39:50	0:08:28	9:04:25
50000	1000	3200	1192	0:00:09	2:12:14	0:08:34	12:07:04
50000	1000	4000	1490	0:00:08	2:46:27	0:09:52	15:13:18
50000	1000	4800	1788	0:00:21	3:24:38	0:13:28	18:11:34
50000	1000	5600	2086	0:00:10	3:50:29	0:14:55	21:25:49
50000	1000	6400	2384	0:00:06	4:26:31	0:16:11	1 day 0:27:56
50000	1000	7200	2682	0:00:11	4:56:57	0:17:11	1 day 3:34:19

SVM with homogeneous data

N	F	Cores	Space(GB)	TWalltime	TCompute	CWalltime	CCompute
50000	20	48	0	0:00:01	0:00:46	0:02:45	2:01:12
50000	20	96	0	0:00:01	0:01:32	0:02:47	4:03:05
50000	20	144	1	0:00:02	0:02:19	0:02:49	5:58:08
50000	20	192	1	0:00:02	0:03:06	0:02:45	7:56:14
50000	20	240	1	0:00:02	0:03:53	0:02:51	9:54:05
50000	50	48	0	0:00:03	0:01:23	0:05:14	3:44:06
50000	50	96	1	0:00:03	0:02:47	0:05:19	7:26:30
50000	50	144	2	0:00:03	0:04:11	0:05:25	11:07:51
50000	50	192	3	0:00:07	0:05:38	0:05:25	14:54:03
50000	50	240	4	0:00:03	0:07:00	0:05:25	18:26:10
50000	100	48	1	0:00:05	0:02:20	0:09:28	6:25:55
50000	100	96	3	0:00:05	0:04:40	0:09:56	12:49:20
50000	100	144	5	0:00:05	0:07:04	0:09:45	19:09:22
50000	100	192	7	0:00:06	0:09:25	0:09:53	1 day 1:27:47
50000	100	240	8	0:00:05	0:11:46	0:10:06	1 day 8:08:18

Star data

Cores	TWalltime	TCompute	CWalltime	CCompute
2500	0:01:06	11:35:25	0:24:39	31 days 19:59:13
3000	0:00:49	12:10:33	0:21:43	32 days 2:44:11
3500	0:00:50	12:17:27	0:17:01	30 days 7:56:19
4000	0:00:45	12:38:24	0:29:53	40 days 13:38:19