Reduction Techniques for Training Support Vector Machines

Kuan-ming Lin

Dept. Computer Science & Information Engineering

National Taiwan University

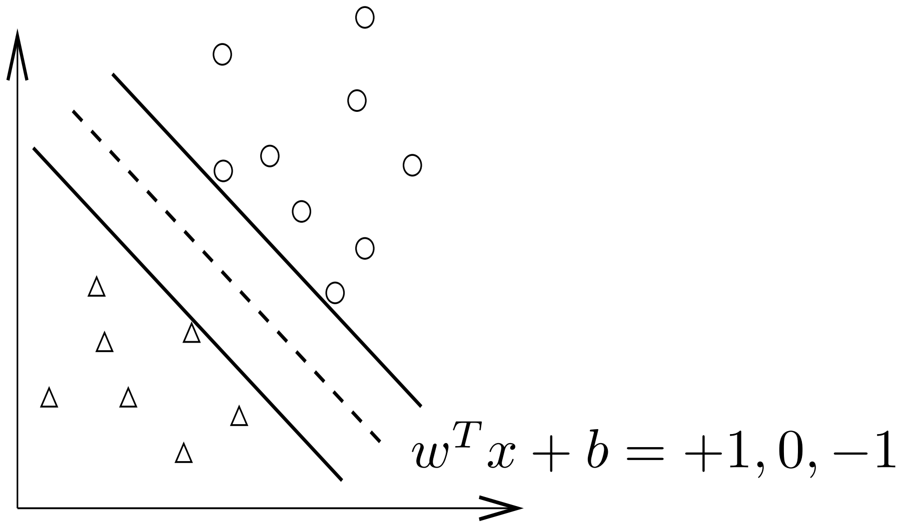
Outline

- Reduced support vector machines (RSVM) analysis
- RSVM implementations
- Performance of RSVM
- Study on incomplete Cholesky factorization (ICF)
- Using ICF kernel approximation technique for SVM (ICFSVM)
- Performance of ICFSVM

Support Vector Machines

- A promising method for data classifications
- Training and testing
- Training vectors : $x_i, i = 1, \dots, l$
- Consider examples with two classes:

$$y_i = \begin{cases} 1 & \text{if } x_i \text{ in class 1} \\ -1 & \text{if } x_i \text{ in class 2} \end{cases}$$



- Variables: w and b : Need to know coefficients of a plane
- Decision function $w^T x + b$, x: test data

SVM Formulation

- Maximize the margin $2/\|w\| \equiv$ Minimize $w^T w/2$
- Apply nonlinear mapping ϕ for training data
- Avoid overfitting for training data: allow training error ξ
- A standard problem [Cortes and Vapnik, 1995]:

$$
\begin{aligned}
\min_{w,b,\xi} \quad & \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i \\
& y_i(w^T\phi(x_i)+b) \geq 1-\xi_i, \\
& \xi_i \geq 0,\ i=1,\ldots,l
\end{aligned}
$$

The Dual Problem

- w: a vector in an infinite dimensional space
- Solve the SVM dual problem:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \\ & 0 \le \alpha_i \le C, i = 1, \ldots, l \\ & y^T\alpha = 0 \end{aligned}$$

e: vector of all ones, $Q_{ij} = y_i y_j \phi(x_i)^T \phi(x_j)$.

- At optimal solution $w = \sum_{i=1}^{l} \alpha_i y_i \phi(x_i)$

Large-scale SVM Problems

- Q fully dense, cannot be saved in memory: traditional optimization methods not usable
- Decomposition methods: currently major approach
  - iteratively solve smaller problems by fixing most variables
  - slow convergence for huge problems with many support vectors

- Reduction techniques:
  - alter the standard SVM formulation
  - reduce the size of Q and solve the reduced problem
- For how large problems is reduction better?

  Performance not fully studied before
  - testing accuracy: compare with standard SVM
  - training time: compare with decomposition methods

The Reduced Support Vector Machine

- Proposed in [Lee and Mangasarian, 2001]
- Start from a variant of SVM:

$$\begin{aligned}\min_{w,b,\xi} \quad & \frac{1}{2}(w^T w + b^2) + C\sum_{i=1}^{l} \xi_i^2 \\ & y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i,\ i = 1, \dots, l\end{aligned}$$

- Let w be $\sum_{i=1}^{l} \alpha_i y_i \phi(x_i)$ (here α not dual variable):

$$\begin{aligned}\min_{\alpha,b,\xi} \quad & \frac{1}{2}(\alpha^T Q\alpha + b^2) + C\sum_{i=1}^{l} \xi_i^2 \\ & Q\alpha + by \geq e - \xi\end{aligned}$$

- RSVM randomly selects a subset R of m samples as support vectors: $w = \sum_{i \in R} \alpha_i y_i \phi(x_i)$

  let $\bar{\alpha} \equiv \alpha_R$

$$\min_{\bar{\alpha}, b, \xi} \quad \frac{1}{2}(\bar{\alpha}^T Q_{RR} \bar{\alpha} + b^2) + C \sum_{i=1}^{l} \xi_i^2$$

$$Q_{:,R} \bar{\alpha} + by \geq e - \xi$$

- Simplify $1/2 \bar{\alpha}^T Q_{RR} \bar{\alpha}$ to $1/2 \bar{\alpha}^T \bar{\alpha}$

- Absorb b by $\widetilde{Q} \equiv \begin{bmatrix} Q_{:,R} & y \end{bmatrix}, \widetilde{\alpha} \equiv \begin{bmatrix} \bar{\alpha} \\ b \end{bmatrix}$
- The formulation of RSVM:

$$\begin{aligned} \min_{\widetilde{\alpha},\xi} \quad & \frac{1}{2}\widetilde{\alpha}^T\widetilde{\alpha} + C\sum_{i=1}^{l}\xi_i^2 \\ & \widetilde{Q}\widetilde{\alpha} \geq e - \xi \end{aligned}$$

- We find it similar to radical basis function networks
  - comparisons of RBF networks with SVM was done

How to Solve RSVM

- Smooth SVM (SSVM) in [Lee and Mangasarian, 2001]
- Transform RSVM to an unconstrained problem:

$$\min_{\tilde{\alpha}} \quad \frac{1}{2}\tilde{\alpha}^T\tilde{\alpha} + C\sum_{i=1}^{l}((e - \tilde{Q}\tilde{\alpha})_i)_+^2$$

- $(.)_+ \equiv max(.,0)$ not differentiable
- Approximate $(t)_+$ by $P_\beta(t) \equiv t + \beta^{-1}\log(1+\exp(-\beta t))$:
  Differentiable, Newton's method can be used
- Each iteration $O(lm^2)$ time for Hessian (2nd derivatives)

RSVM is Already in the Form of Linear SVM

- Linear SVM primal form:

$$\min_{w,\xi} \quad \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i^2$$
$$YXw \geq e - \xi$$

- Formulation same with RSVM:

$$\min_{\tilde{\alpha},\xi} \quad \frac{1}{2}\tilde{\alpha}^T\tilde{\alpha} + C\sum_{i=1}^{l}\xi_i^2$$
$$\tilde{Q}\tilde{\alpha} \geq e - \xi$$

- Dimension of $\tilde{\alpha}$ is $(m+1) \ll l$: proper methods exist

Use Least Square SVM

- Proposed in [Suykens and Vandewalle, 1999]
  - Change $\tilde{Q}\tilde{\alpha} \geq e - \xi$ to equality $\tilde{Q}\tilde{\alpha} = e - \xi$
- ξ is represented by $\tilde{\alpha}$:

$$\min_{\tilde{\alpha}} \quad f(\tilde{\alpha}) = \quad \frac{1}{2}\tilde{\alpha}^T\tilde{\alpha} + C\sum_{i=1}^{l}(e - \tilde{Q}\tilde{\alpha})_i^2$$

- Quadratic unconstrained, equivalent to a linear system:

$$(\tilde{Q}^T\tilde{Q} + \frac{I}{2C})\tilde{\alpha} = \tilde{Q}^T e$$

- Main cost is $O(lm^2)$ to calculate $\tilde{Q}^T\tilde{Q}$

Use Decomposition

- The dual form of RSVM:

$$\min_{\alpha} \quad \frac{1}{2}\alpha^T(\widetilde{Q}\widetilde{Q}^T + \frac{I}{2C})\alpha - e^T\alpha$$
$$0 \le \alpha_i, i = 1, \ldots, l$$

  - primal RSVM solution $\widetilde{\alpha} = \widetilde{Q}^T\alpha$

- Each iteration a working set of size q is to be modified
- Main cost is calculating $Q\Delta\alpha$: $O(lqm)$ for $O(m)$ kernel
- Speedup for linear kernel and RSVM: $Q\Delta\alpha = \widetilde{Q}(\widetilde{Q}^T\Delta\alpha)$
  $O(mq) + O(lm) = O(lm)$ operations, q times faster
- Used in software SVM^{light} and BSVM

Use Lagrangian SVM

- Proposed in [Mangasarian and Musicant, 2001]
- Consider optimality condition of dual RSVM:
  $H\alpha - e \geq 0, \alpha \geq 0, (H\alpha - e)^T\alpha = 0$ with $H \equiv \widetilde{Q}\widetilde{Q}^T + \frac{I}{2C}$
- Equivalent to $H\alpha - e = (H\alpha - e - \beta\alpha)_+, \forall\beta > 0$:
  apply fixed-point iterations (each step $O(lm)$ time)
  $\alpha^{k+1} = H^{-1}(e + (H\alpha^k - e - \beta\alpha^k)_+)$
- Can obtain H^{-1} by SMW identity only for $m \ll l$:
  $H^{-1} = (\frac{I}{2C} + \widetilde{Q}\widetilde{Q}^T)^{-1} = 2C(I - \widetilde{Q}(\frac{I}{2C} + \widetilde{Q}^T\widetilde{Q})^{-1}\widetilde{Q}^T)$

Implementation Issues

- Stopping criteria for iterative methods
  - RSVM form different from SVM
  - we choose as close criteria as possible
- Multi-class problems: we use one-against-one
  - $k(k-1)/2$ classifiers for k classes where each one trains data from two classes, when testing they vote
  - suggested in surveys of multi-class SVM, LS-SVM

Problems for Experiments

Problem	#training data	#testing data	#class	#attribute
dna	2000	1300	3	180
satimage	4435	2000	6	36
letter	15000	5000	26	16
shuttle	43500	14500	7	9
mnist	21000	49000	10	780
ijcnn1	49990	45495	2	22
protein	17766	6621	3	357

- Scaling
- Don't use mnist original 60000 training and 10000 testing because too much training time

Settings

- Decomposition solvers for SVM: libsvm and libsvm-q
- $m = 0.1l$ in most cases
- RBF kernel used, model selection for C and γ
  - 70%-30% hold-out, 15×15 grid search
- ATLAS to speed up matrix operations
- Caching and shrinking for decomposition methods

Table 1: A comparison on RSVM: testing accuracy

	SVM		RSVM							
	libsvm		SSVM		LS-SVM		LSVM		Decomposition	
Problem	C,γ	rate	C,γ	rate	C,γ	rate	C,γ	rate	C,γ	rate
dna	$2^{4},2^{-6}$	95.447	$2^{12},2^{-10}$	92.833	$2^{4},2^{-6}$	92.327	$2^{5},2^{-7}$	93.002	$2^{9},2^{-6}$	92.327
satimage	$2^{4},2^{0}$	91.3	$2^{12},2^{-1}$	89.8	$2^{12},2^{-3}$	89.9	$2^{2},2^{-1}$	90	$2^{11},2^{-1}$	90
letter	$2^{4},2^{2}$	97.98	$2^{11},2^{-1}$	95.9	$2^{12},2^{-2}$	95.14	$2^{12},2^{-1}$	95.42	$2^{12},2^{-1}$	92.76
shuttle	$2^{11},2^{3}$	99.924	$2^{12},2^{4}$	99.78	$2^{12},2^{4}$	99.58	$2^{10},2^{3}$	99.814	$2^{12},2^{4}$	99.772
mnist	$2^{6},2^{-5}$	97.753	$2^{7},2^{-6}$	96.833	$2^{9},2^{-6}$	96.48	$2^{4},2^{-5}$	96.578	$2^{12},2^{-5}$	96.129
ijcnn1	$2^{1},2^{1}$	98.76	$2^{12},2^{-3}$	95.949	$2^{-2},2^{-2}$	91.676	$2^{12},2^{-3}$	96.813	$2^{12},2^{-1}$	96.11
protein	$2^{1},2^{-3}$	69.97	$2^{1},2^{-5}$	65.957	$2^{2},2^{-6}$	66.244	$2^{0},2^{-5}$	65.957	$2^{11},2^{-6}$	66.138

- libsvm-q very close to libsvm, not listed here

Table 2: A comparison on RSVM: number of support vectors

	SVM		RSVM			
	libsvm	libsvm-q	SSVM	LS-SVM	LSVM	Decomposition
Problem	#SV		#SV (all same)			
dna	973	1130	372			
satimage	1611	1822	1826			
letter	8931	8055	13928			
shuttle	285	652	4982			
mnist	8333	8364	12874			
ijcnn1	4555	9766	200			
protein	14770	16192	596			

Table 3: A comparison on RSVM: training time and testing time (in seconds)

	SVM				RSVM				
	libsvm		libsvm-q		SSVM	LS-SVM	LSVM	Decomposition	
Problem	training	testing	training	testing	training	training	training	training	testing
dna	7.09	4.65	8.5	5.39	5.04	2.69	23.4	7.59	1.52
satimage	16.21	9.04	19.04	10.21	23.77	11.59	141.17	43.75	11.4
letter	230	89.53	140.14	75.24	193.39	71.06	1846.12	446.04	149.77
shuttle	113	2.11	221.04	3.96	576.1	150.59	3080.56	562.62	74.82
mnist	1265.67	4475.54	1273.29	4470.95	1464.63	939.76	4346.28	1913.86	7836.99
ijcnn1	492.53	264.58	2791.5	572.58	57.87	19.42	436.46	16152.54	6.36
protein	1875.9	687.9	9862.25	808.68	84.21	64.6	129.47	833.35	35

Observations

- Accuracy: all RSVM implementations lower than SVM
- LS-SVM a little lower among RSVM implementations
- Optimal models for RSVM have much larger C
- For median-sized problems RSVM not much faster
- RSVM is much faster for ijcnn1 and protein
  - larger problem or many support vectors for SVM
  - m is set small
  - LS-SVM fastest

Incomplete Cholesky Factorization

- Find lower triangular V: VV^T approximates a matrix
- Primarily used for conjugate gradient methods
- Used for SVM in [Fine and Scheinberg, 2001]
  - motivation: to solve SVM by interior point method, low-rank representation $Q \sim VV^T$ needed
  - factorize Q
    * large dense, entries calculated when needed
    * only some ICF algorithms are suitable

ICF Algorithms

- Based on a columnwize Cholesky factorization method:

$$\begin{bmatrix} \alpha & v^T \\ v & B \end{bmatrix} = \begin{bmatrix} \sqrt{\alpha} & 0 \\ \frac{v}{\sqrt{\alpha}} & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & B - \frac{vv^T}{\alpha} \end{bmatrix} \begin{bmatrix} \sqrt{\alpha} & \frac{v^T}{\sqrt{\alpha}} \\ 0 & I \end{bmatrix}$$

- 1st ICF algorithm in [Lin and Saigal, 2000]
  - stores largest m values in each column of V
  - may fail: add βI and restart
- 2nd ICF algorithm in [Fine and Scheinberg, 2001]
  - early stop, fewer columns of Cholesky factorization
  - also uses symmetric pivoting

The Approximate Problem is in Linear SVM Form

- Linear SVM dual form:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T (YX(YX)^T + yy^T)\alpha - e^T\alpha \\ & 0 \le \alpha_i \le C, i = 1, \ldots, l \end{aligned}$$

- Approximate dual form by ICF, called ICFSVM:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T (VV^T + yy^T)\alpha - e^T\alpha \\ & 0 \le \alpha_i \le C, i = 1, \ldots, l \end{aligned}$$

Implementations

- Solving primal (SSVM,LS-SVM) versus solving dual (LSVM, decomposition)
- ICFSVM in dual form: use decomposition to implement
- Should we solve the corresponding primal?

$$\begin{array}{ll} \displaystyle\min_{\widetilde{w},\xi} & \displaystyle\frac{1}{2}\widetilde{w}^T\widetilde{w} + C(\sum_{i=1}^{l}\xi_i) \\ \text{subject to} & V\widetilde{w} \geq e - \xi, \xi \geq 0 \end{array}$$

- V not meaningful in primal: solution cannot be used

Experimental Results

- $m = 0.1l$ in most cases, same way with RSVM

Table 4: A comparison on ICFSVM: testing accuracy

	SVM		RSVM		ICFSVM(decomposition implementation)					
	libsvm		Decomposition		1st ICF		2nd ICF		2nd ICF+retrain	
Problem	C, γ	rate	C, γ	rate	C, γ	rate	C, γ	rate	C, γ	rate
dna	$2^4, 2^{-6}$	95.447	$2^9, 2^{-6}$	92.327	$2^{-1}, 2^{-4}$	66.273	$2^0, 2^{-9}$	92.917	$2^2, 2^{-4}$	87.436
satimage	$2^4, 2^0$	91.3	$2^{11}, 2^{-1}$	90	$2^{-1}, 2^1$	87.2	$2^6, 2^{-5}$	89.2	$2^1, 2^1$	72.65
letter	$2^4, 2^2$	97.98	$2^{12}, 2^{-1}$	92.76	$2^2, 2^2$	96.66	$2^5, 2^{-2}$	96.16	$2^0, 2^2$	94.06
shuttle	$2^{11}, 2^3$	99.924	$2^{12}, 2^4$	99.772	$2^{-1}, 2^4$	94.4	$2^9, 2^4$	99.862	$2^0, 2^{-1}$	94.331
mnist	$2^6, 2^{-5}$	97.753	$2^{12}, 2^{-5}$	96.129	N/A	N/A	$2^1, 2^{-7}$	96.045	$2^1, 2^{-5}$	95.259
ijcnn1	$2^1, 2^1$	98.76	$2^{12}, 2^{-1}$	96.11	$2^{-2}, 2^{-10}$	90.185	$2^9, 2^{-5}$	92.274	$2^{-2}, 2^0$	91.711
protein	$2^1, 2^{-3}$	69.97	$2^{11}, 2^{-6}$	66.138	N/A	N/A	$2^{-1}, 2^{-6}$	65.398	$2^{-2}, 2^{-5}$	65.926

N/A: training time too large to apply the model selection

Table 5: A comparison on ICFSVM: number of support vectors

	SVM	RSVM	ICFSVM		
	libsvm	Decomposition	1st ICF	2nd ICF	2nd ICF+retrain
Problem	#SV				
dna	973	372	1688	**1389**	1588
satimage	1611	1826	4022	**1187**	1507
letter	8931	13928	12844	**5390**	8953
shuttle	285	4982	43026	**308**	3714
mnist	8333	12874	N/A	**5295**	5938
ijcnn1	4555	200	49485	**4507**	8731
protein	14770	596	N/A	**15049**	15512

N/A: training time too large to apply the model selection

Table 6: A comparison on ICFSVM: training time and ICF time (in seconds)

	SVM	RSVM	ICFSVM					
	libsvm	Decomposition	1st ICF		2nd ICF		2nd ICF+retrain	
Problem	training	training	training	ICF	training	ICF	training	ICF
dna	7.09	7.59	440.41	427.18	9.62	5.45	33.77	5.52
satimage	16.21	43.75	558.23	467.48	48.49	28.37	61.59	28.32
letter	230	446.04	3565.31	2857.95	484.59	222.4	635.41	221.93
shuttle	113	562.62	70207.76	13948.14	1251.17	1184.63	1811.6	1265.51
mnist	1265.67	1913.86	N/A	N/A	2585.13	2021.64	2565.08	1866.9
ijcnn1	492.53	16152.54	21059.3	4680.63	5463.8	103.97	1579.73	102.52
protein	1875.9	833.35	N/A	N/A	217.53	92.52	3462.57	110.54

N/A: training time too large to apply the model selection

Discussions and Conclusions

- ICFSVM accuracy like RSVM, lower than SVM

  Used if decomposition for SVM cannot afford
- ICFSVM optimal models have smaller C than RSVM
- Support vectors of ICFSVM as sparse as SVM
- ICFSVM not faster than RSVM: ICF time too much
- Retrain SV from ICFSVM by SVM: not good
- First algorithm strange: ICF may not close