Working Set Selection Using Second Order Information for Training SVM

Chih-Jen Lin
Department of Computer Science
National Taiwan University

Joint work with Rong-En Fan and Pai-Hsuen Chen

Talk at NIPS 2005 Workshop on Large Scale Kernel Machines

Outline

- Large dense quadratic programming in SVM
- Decomposition methods and working set selections
- A new selection based on second order information
- Results and analysis
- This work appears in JMLR 2005

SVM Dual Optimization Problem

- Large dense quadratic problem

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

- l: # of training data
- Q: l by l fully dense matrix
- $y_i = \pm 1$
- $\mathbf{e} = [1, \ldots, 1]^T$
- Difficult as Q is fully dense in general

- Do we really need to solve the dual?

  Maybe not. Sometimes data too large to do so
- Approximating either from primal or dual side
- However, in certain situations we still hope to solve it

  This talk: a faster algorithm and implementation

Decomposition Methods

- Working on a few variable each time
- Similar to coordinate-wise minimization
- Working set B, $N = \{1, \ldots, l\} \backslash B$ fixed

  Size of B usually $<= 100$
- Sub-problem in each iteration:

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}_B} \quad & \frac{1}{2} \begin{bmatrix} \boldsymbol{\alpha}_B^T & (\boldsymbol{\alpha}_N^k)^T \end{bmatrix} \begin{bmatrix} Q_{BB} & Q_{BN} \\ Q_{NB} & Q_{NN} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_B \\ \boldsymbol{\alpha}_N^k \end{bmatrix} - \\
& \begin{bmatrix} \mathbf{e}_B^T & (\mathbf{e}_N^k)^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_B \\ \boldsymbol{\alpha}_N^k \end{bmatrix} \\
\text{subject to} \quad & 0 \le (\boldsymbol{\alpha}_B)_t \le C, t = 1, \ldots, q, \ \mathbf{y}_B^T \boldsymbol{\alpha}_B = -\mathbf{y}_N^T \boldsymbol{\alpha}_N^k
\end{aligned}
$$

Sequential Minimal Optimization (SMO)

- Consider $B = \{i, j\}$; that is, $|B| = 2$ (Platt, 1998)

  Extreme of decomposition methods
- Sub-problem analytically solved; no need to use optimization software

$$
\begin{aligned}
\min_{\alpha_i, \alpha_j} \quad & \frac{1}{2} \begin{bmatrix} \alpha_i & \alpha_j \end{bmatrix} \begin{bmatrix} Q_{ii} & Q_{ij} \\ Q_{ij} & Q_{jj} \end{bmatrix} \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} + (Q_{BN}\boldsymbol{\alpha}_N^k - \mathbf{e}_B)^T \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} \\
\text{s.t.} \quad & 0 \le \alpha_i, \alpha_j \le C, \\
& y_i\alpha_i + y_j\alpha_j = -\mathbf{y}_N^T \boldsymbol{\alpha}_N^k,
\end{aligned}
$$

- This work focuses on selecting two elements

Existing Selection by Gradient

- Let $\mathbf{d} \equiv [\mathbf{d}_B, \mathbf{0}_N]$. Minimizing

$$\begin{aligned} f(\boldsymbol{\alpha}^k + \mathbf{d}) &\approx f(\boldsymbol{\alpha}^k) + \nabla f(\boldsymbol{\alpha}^k)^T \mathbf{d} \\ &= f(\boldsymbol{\alpha}^k) + \nabla f(\boldsymbol{\alpha}^k)_B^T \mathbf{d}_B. \end{aligned}$$

- Solve

$$\begin{aligned} \min_{\mathbf{d}_B} \quad & \nabla f(\boldsymbol{\alpha}^k)_B^T \mathbf{d}_B \\ \text{subject to} \quad & \mathbf{y}_B^T \mathbf{d}_B = 0, \\ & d_t \geq 0, \text{ if } \alpha_t^k = 0, t \in B, & \text{(1a)} \\ & d_t \leq 0, \text{ if } \alpha_t^k = C, t \in B, & \text{(1b)} \\ & -1 \leq d_t \leq 1, t \in B \\ & |B| = 2 \end{aligned}$$

- First considered in (Joachims, 1998)
- $0 \leq \alpha_t \leq C$ leads to (1a) and (1b).

$$
\begin{aligned}
0 \leq \alpha_t^k + d_t \quad &\Rightarrow \quad d_t \geq 0, \text{ if } \alpha_t^k = 0, \\
\alpha_t^k + d_t \leq C \quad &\Rightarrow \quad d_t \leq 0, \text{ if } \alpha_t^k = C
\end{aligned}
$$

$\boldsymbol{\alpha} + \mathbf{d}$ may not be feasible. OK for finding working sets

- $-1 \leq d_t \leq 1, t \in B$ avoid $-\infty$ objective value

- Rewritten as checking first order approximation at different sub-problems of B

$$\{i,j\} = \arg \min_{B:|B|=2} \text{Sub}(B),$$

where

$$\begin{aligned} \text{Sub}(B) \equiv \min_{\mathbf{d}_B} \quad & \nabla f(\boldsymbol{\alpha}^k)_B^T \mathbf{d}_B \\ \text{subject to} \quad & \mathbf{y}_B^T \mathbf{d}_B = 0, \\ & d_t \geq 0, \text{ if } \alpha_t^k = 0, t \in B, \\ & d_t \leq 0, \text{ if } \alpha_t^k = C, t \in B, \\ & -1 \leq d_t \leq 1, t \in B. \end{aligned}$$

- Checking all $\binom{l}{2}$ possible B's?

Solution of Using Gradient Information

- $O(l)$ procedure

$$i \in \arg \max_{t \in I_{\text{up}}(\boldsymbol{\alpha}^k)} -y_t \nabla f(\boldsymbol{\alpha}^k)_t,$$

$$j \in \arg \min_{t \in I_{\text{low}}(\boldsymbol{\alpha}^k)} -y_t \nabla f(\boldsymbol{\alpha}^k)_t,$$

where

$$I_{\text{up}}(\boldsymbol{\alpha}) \equiv \{t \mid \alpha_t < C, y_t = 1 \text{ or } \alpha_t > 0, y_t = -1\}, \text{ and}$$

$$I_{\text{low}}(\boldsymbol{\alpha}) \equiv \{t \mid \alpha_t < C, y_t = -1 \text{ or } \alpha_t > 0, y_t = 1\}.$$

- This usually called maximal violating pair

Better Working Set Selection

- Difficult: # iter $\searrow$ but cost per iter $\nearrow$

  May not imply shorter training time

- A selection by second order information (Fan et al., 2005)

  As f is a quadratic,

$$
\begin{aligned}
f(\boldsymbol{\alpha}^k + \mathbf{d}) &= f(\boldsymbol{\alpha}^k) + \nabla f(\boldsymbol{\alpha}^k)^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla^2 f(\boldsymbol{\alpha}^k) \mathbf{d} \\
&= f(\boldsymbol{\alpha}^k) + \nabla f(\boldsymbol{\alpha}^k)_B^T \mathbf{d}_B + \frac{1}{2} \mathbf{d}_B^T \nabla^2 f(\boldsymbol{\alpha}^k)_{BB} \mathbf{d}_B
\end{aligned}
$$

Selection by Second-Order Information

- Using second order information

$$\min_{B:|B|=2} \text{Sub}(B),$$

$$\begin{aligned}
\text{Sub}(B) \equiv \min_{\mathbf{d}_B} \quad & \frac{1}{2}\mathbf{d}_B^T \nabla^2 f(\boldsymbol{\alpha}^k)_{BB}\mathbf{d}_B + \nabla f(\boldsymbol{\alpha}^k)_B^T \mathbf{d}_B \\
\text{subject to} \quad & \mathbf{y}_B^T \mathbf{d}_B = 0, \\
& d_t \geq 0, \text{ if } \alpha_t^k = 0, t \in B, \\
& d_t \leq 0, \text{ if } \alpha_t^k = C, t \in B.
\end{aligned}$$

- $-1 \leq d_t \leq 1, t \in B$ not needed if Q_{BB} PD
- Too expensive to check $\binom{l}{2}$ sets

- A heuristic

1. Select
$$i \in \arg\max_t \{-y_t \nabla f(\boldsymbol{\alpha}^k)_t \mid t \in I_{\text{up}}(\boldsymbol{\alpha}^k)\}.$$
2. Select
$$\begin{aligned} j \in \arg\min_t \{\text{Sub}(\{i,t\}) \mid t \in I_{\text{low}}(\boldsymbol{\alpha}^k),\\ -y_t \nabla f(\boldsymbol{\alpha}^k)_t < -y_i \nabla f(\boldsymbol{\alpha}^k)_i\}. \end{aligned}$$
3. Return $B = \{i, j\}$.

- The same i as using the gradient information

  Check only $O(l)$ B's to decide j

- Sub$(\{i,t\})$ can be easily solved

  If $K_{ii} + K_{jj} - 2K_{ij} > 0$,

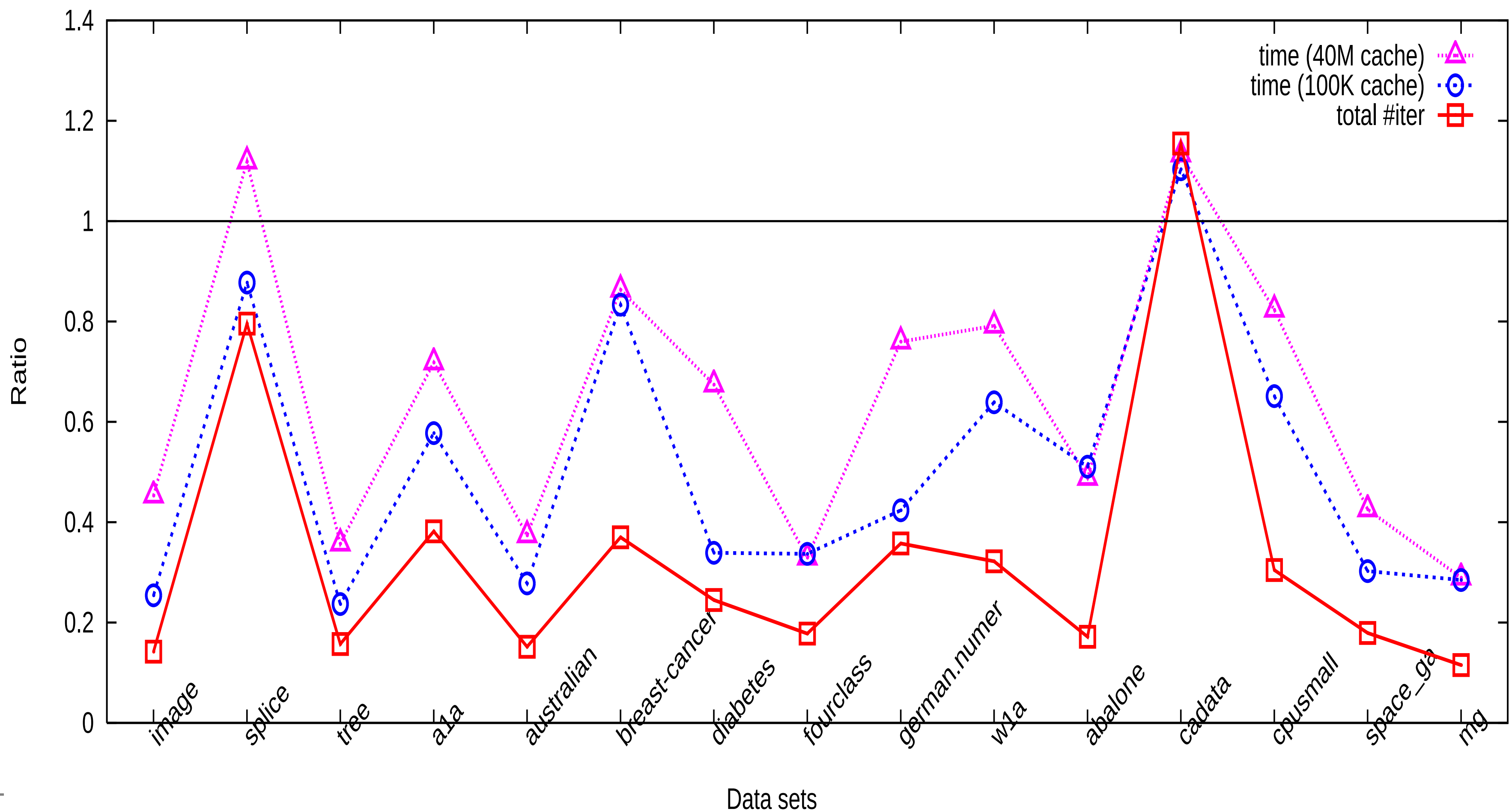
$$\text{Sub}(\{i,t\}) = -\frac{(-y_i \nabla f(\boldsymbol{\alpha}^k)_i + y_t \nabla f(\boldsymbol{\alpha}^k)_t)^2}{2(K_{ii} + K_{tt} - 2K_{it})}$$

- Convergence established in (Fan et al., 2005)

  Details not shown here

Comparison of Two Selections

- Iteration and time ratio between using second-order information and maximal violating pair



- A complete comparison is not easy

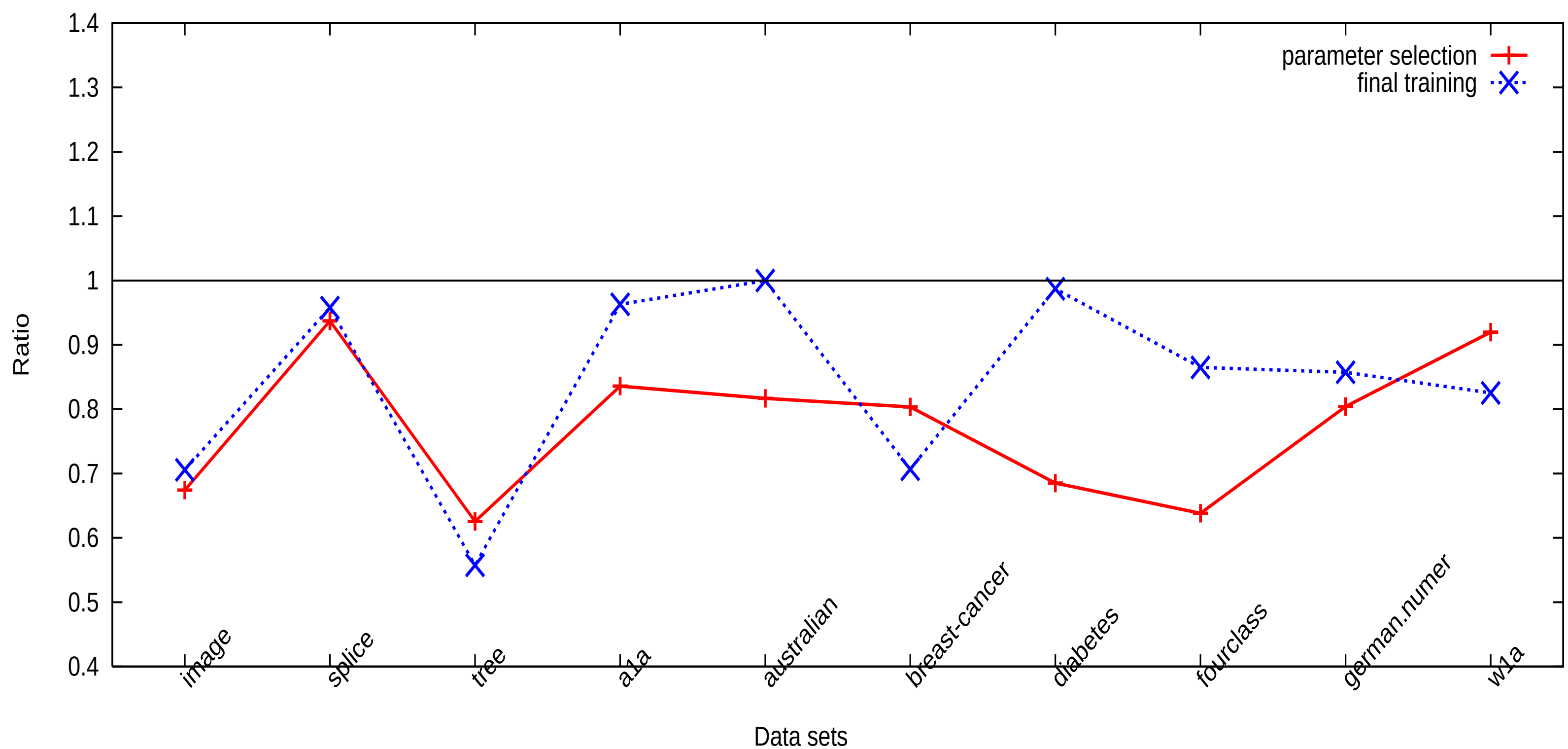
  Try enough data sets

  Consider parameter selection
- Details not shown here

More about Second-Order Selection

- What if we check all $\binom{l}{2}$ sets

  Iteration ratio between checking all and checking $O(l)$:



- Fewer iterations, but ratio (0.7 to 0.8) not enough to justify the higher cost per iteration

Why not Keeping Feasibility?

$$\min_{\mathbf{d}_B} \qquad \frac{1}{2}\mathbf{d}_B^T \nabla^2 f(\boldsymbol{\alpha}^k)_{BB}\mathbf{d}_B + \nabla f(\boldsymbol{\alpha}^k)_B^T \mathbf{d}_B$$

- Two types of constraints:

$$\mathbf{y}_B^T\mathbf{d}_B = 0, \qquad\qquad \mathbf{y}_B^T\mathbf{d}_B = 0,$$

$$d_t \geq 0, \text{ if } \alpha_t^k = 0, t \in B, \qquad\qquad 0 \leq \alpha^k + d_t \leq C, t \in B$$

$$d_t \leq 0, \text{ if } \alpha_t^k = C, t \in B$$

- Related work (Lai et al., 2003a,b)
  - Heuristically select some pairs
  - Check function reduction while keeping feasibility
- Higher cost in selecting working sets
- We proved: at final iterations two are indeed the same

Conclusions

- Finding better working sets for SVM decomposition methods is difficult
- We proposed one based on second order information
- Results better than the commonly used selection from first order information
- Implementation in LIBSVM (after version 2.8)

  `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

  Replacing the maximal violating pair selection