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

\[
\begin{align*}
\min_{\alpha} & \quad \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \\
\text{subject to} & \quad 0 \leq \alpha_i \leq C, i = 1, \ldots, l \\
& \quad y^T \alpha = 0,
\end{align*}
\]

- \( l \): # of training data
- \( Q \): \( l \) by \( l \) fully dense matrix
- \( y_i = \pm 1 \)
- \( e = [1, \ldots, 1]^T \)
- 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 $\leq 100$

Sub-problem in each iteration:

$$\min_{\alpha_B} \frac{1}{2} \left[ \begin{array}{c} \alpha_B^T \\ (\alpha_N^k)^T \end{array} \right] \left[ \begin{array}{cc} Q_{BB} & Q_{BN} \\ Q_{NB} & Q_{NN} \end{array} \right] \left[ \begin{array}{c} \alpha_B \\ \alpha_N^k \end{array} \right] -$$

$$\left[ e_B^T \\ (e_N^k)^T \right] \left[ \begin{array}{c} \alpha_B \\ \alpha_N^k \end{array} \right]$$

subject to $0 \leq (\alpha_B)_t \leq C, t = 1, \ldots, q, \ y_B^T \alpha_B = -y_N^T \alpha_N^k$
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

\[
\begin{align*}
\min_{\alpha_i, \alpha_j} & \quad \frac{1}{2} \begin{bmatrix} \alpha_i & \alpha_j \end{bmatrix} \begin{bmatrix} Q_{ii} & Q_{ij} \\ Q_{ij} & Q_{jj} \end{bmatrix} \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} + (Q_{BN} \alpha_N^k - e_B)^T \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} \\
\text{s.t.} & \quad 0 \leq \alpha_i, \alpha_j \leq C, \\
& \quad y_i \alpha_i + y_j \alpha_j = -y_N^T \alpha_N^k,
\end{align*}
\]

- This work focuses on selecting two elements
Existing Selection by Gradient

Let $d \equiv [d_B, 0_N]$. Minimizing

$$f(\alpha^k + d) \approx f(\alpha^k) + \nabla f(\alpha^k)^T d$$
$$= f(\alpha^k) + \nabla f(\alpha^k)_B^T d_B.$$ 

Solve

$$\min_{d_B} \nabla f(\alpha^k)_B^T d_B$$

subject to

$$y_B^T d_B = 0,$$
$$d_t \geq 0, \text{ if } \alpha^k_t = 0, t \in B,$$  
$$d_t \leq 0, \text{ if } \alpha^k_t = C, t \in B,$$  
$$-1 \leq d_t \leq 1, t \in B$$  
$$|B| = 2.$$
First considered in (Joachims, 1998)

$0 \leq \alpha_t \leq C$ leads to (1a) and (1b).

$0 \leq \alpha^k_t + d_t \Rightarrow d_t \geq 0$, if $\alpha^k_t = 0$,

$\alpha^k_t + d_t \leq C \Rightarrow d_t \leq 0$, if $\alpha^k_t = C$

$\alpha + d$ 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

\[
\text{Sub}(B) \equiv \min_{d_B} \nabla f(\alpha^k)_B^T d_B
\]

subject to

\[ y_B^T d_B = 0, \]
\[ d_t \geq 0, \text{ if } \alpha^k_t = 0, t \in B, \]
\[ d_t \leq 0, \text{ if } \alpha^k_t = C, t \in B, \]
\[ -1 \leq d_t \leq 1, t \in B. \]

Checking all $\binom{l}{2}$ possible $B$'s?
Solution of Using Gradient Information

- $O(l)$ procedure

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

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

where

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

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

- This usually called maximal violating pair
Better Working Set Selection

- Difficult: # iter ↓ but cost per iter ↑

May not imply shorter training time

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

As $f$ is a quadratic,

$$f(\alpha^k + d) = f(\alpha^k) + \nabla f(\alpha^k)^T d + \frac{1}{2}d^T \nabla^2 f(\alpha^k) d$$

$$= f(\alpha^k) + \nabla f(\alpha^k)_B d_B + \frac{1}{2}d_B^T \nabla^2 f(\alpha^k)_B d_B$$
Using second order information

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

\[
\text{Sub}(B) \equiv \min_{d_B} \frac{1}{2} d_B^T \nabla^2 f(\alpha^k) d_B + \nabla f(\alpha^k)^T d_B
\]

subject to

\[
y_B^T d_B = 0,
\]

\[
d_t \geq 0, \text{ if } \alpha^k_t = 0, t \in B,
\]

\[
d_t \leq 0, \text{ if } \alpha^k_t = C, t \in B.
\]

\(-1 \leq d_t \leq 1, t \in B \) not needed if \( Q_{BB} \) PD

Too expensive to check \( \binom{l}{2} \) sets
A heuristic

1. Select

\[ i \in \text{arg max}_t \{ -y_t \nabla f(\alpha^k)_t \mid t \in I_{\text{up}}(\alpha^k) \}. \]

2. Select

\[ j \in \text{arg min}_t \{ \text{Sub}(\{i, t\}) \mid t \in I_{\text{low}}(\alpha^k), -y_t \nabla f(\alpha^k)_t < -y_i \nabla f(\alpha^k)_i \}. \]

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 \( K_{ii} + K_{jj} - 2K_{ij} > 0 \),

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

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?

\[
\min_{d_B} \quad \frac{1}{2} d_B^T \nabla^2 f(\alpha^k)_{BB} d_B + \nabla f(\alpha^k)^T B d_B
\]

- **Two** types of constraints:

  \[
y_B^T d_B = 0,
  \]

  \[
d_t \geq 0, \text{ if } \alpha^k_t = 0, t \in B,
  \]

  \[
d_t \leq 0, \text{ if } \alpha^k_t = C, t \in B
  \]

- 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