ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

On related violating pairs for working set selection in SMO algorithms

Tobias Glasmachers

Optimization of Adaptive Systems Group
Institut für Neuroinformatik
Ruhr-Universität Bochum - Germany

Abstract. Sequential Minimal Optimization (SMO) is currently the most popular algorithm to solve large quadratic programs for Support Vector Machine (SVM) training. For many variants of this iterative algorithm proofs of convergence to the optimum exist. Nevertheless, to find such proofs for elaborated SMO-type algorithms is challenging in general. We provide a basic tool for such convergence proofs in the context of cache-friendly working set selection. Finally this result is applied to notably simplify the convergence proof of the very efficient Hybrid Maximum Gain algorithm.

1 Introduction

In the standard supervised learning scenario we are given a training dataset $(x_1, y_1), \ldots, (x_\ell, y_\ell)$ of i.i.d. examples consisting of input $x \in X$ and (target) label $y \in Y$. Support Vector Machines (SVMs) are a state of the art machine learning method for the construction of hypotheses $h : X \to Y$ from such data. Given Mercer kernel function $k : X \times X \to \mathbb{R}$ and regularization parameters depending on the type of SVM used, the machine is trained by solving a quadratic program of the form

$$\begin{aligned} \text{maximize} \quad & f(\alpha) = y^T \alpha - \frac{1}{2} \alpha^T K \alpha && (1) \\ \text{s.t.} \quad & \sum_{i=1}^{\ell} \alpha_i = 0 && \text{(equality constraint)} \\ \text{and} \quad & L_i \leq \alpha_i \leq U_i \;\; \forall 1 \leq i \leq \ell && \text{(box constraint)} \end{aligned}$$

for $\alpha \in \mathbb{R}^\ell$. In the case of binary classification the vector $y \in \mathbb{R}^\ell$ is composed of the training labels, with components ± 1. The positive semi definite $\ell \times \ell$-matrix K is the kernel Gram matrix of the training inputs, that is, $K_{ij} = k(x_i, x_j)$. The lower and upper bounds are fixed to $L_i = \min\{0, y_i C\}$ and $U_i = \max\{0, y_i C\}$ with the regularization parameter $C > 0$. Finally, the SVM hypothesis is constructed from the optimal α of problem (1) as $h(x) = \text{sign}\left(\sum_{i=1}^{\ell} \alpha_i k(x, x_i) + b\right)$, where the offset b is obtained from α after the solution of the quadratic program. That is, the main effort for SVM training is to solve the quadratic program (1). Because the dimension ℓ of this problem can be very large this is a challenging task and the usage of standard optimization software is often impractical.

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

The Sequential Minimal Optimization (SMO) algorithm [1] for the solution of quadratic programs is highly adapted to the special form of the constraints of problem (1). It is a special form of the iterative decomposition algorithm. Its basic skeleton is outlined in Algorithm 1.

Algorithm 1: General SMO Algorithm

Input: feasible initial point α, accuracy ε
compute the initial gradient $\nabla f(\alpha) = y - K\alpha$
do
1 select a working set B
2 update α by solving the sub-problem induced by B
3 compute the gradient $\nabla f(\alpha)$ in the new search point
4 check the stopping condition, depending on ε
loop

In each iteration a tuple $B = (i, j)$ with $i, j \in \{1, \ldots, \ell\}$, $i \neq j$, consisting of two variable indices is selected. It is referred to as the working set,[1] because the algorithm restricts itself to these variables in the current iteration. Working set selection is the most crucial part of the algorithm and we will discuss it in detail in the next section.

After the working set is selected the algorithm updates the search point α by optimally solving the problem with the additional constraint that all variables not indexed by the working set are fixed. Then only two variables need to be considered. Together with one equality constraint the resulting problem is one-dimensional and can be solved very efficiently with a truncated Newton step [1]. The gradient in the new search point is obtained as an update of the old gradient in $\mathcal{O}(\ell)$ operations. As a stopping condition the algorithm usually checks whether the Karush-Kuhn-Tucker (KKT) conditions of optimality are fulfilled up to a small predefined constant $\varepsilon > 0$. For a more detailed introduction to the SMO algorithm we refer to the literature [1, 2, 3, 4, 5].

In general the kernel matrix K does not fit into the available working memory. To avoid costly recomputation of matrix entries modern implementations use a cache to store the most recently used rows of the matrix. Together with shrinking [6] this technique is highly efficient.

2 Working Set Selection

The working set selection policy is the key to the performance of the SMO algorithm as well as to its convergence properties. Note that the algorithm produces a sequence of feasible points. Given the initial solution this sequence is completely determined by the sequence of working sets. Thus, the convergence to the optimum as well as the speed of this convergence crucially depend on the working set selection algorithm. Here, we discuss three working set selection strategies. With $B = (i, j)$ the most violating pair (MVP) selects the pair which

[1] To simplify our notation we use ordered tuples instead of sets in this paper.

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

most strongly violates the KKT conditions of optimality (see [2, 5]):

$$i = \arg\max \left\{ \frac{\partial f}{\partial \alpha_n}(\alpha) \;\middle|\; 1 \le n \le \ell \text{ with } \alpha_n < U_n \right\} \tag{2}$$
$$j = \arg\min \left\{ \frac{\partial f}{\partial \alpha_n}(\alpha) \;\middle|\; 1 \le n \le \ell \text{ with } \alpha_n > L_n \right\} \ .$$

Recently MVP was superseded by second order working set selection methods. The basic idea is to directly maximize the functional progress or gain of the SMO step. Let α be the current solution and let α_B be the subsequent search point after a step with working set B. Then, $g_B(\alpha) = f(\alpha_B) - f(\alpha)$ defines this gain. Fan et al. [3] propose to select the second index as

$$j = \arg\max \left\{ \frac{1}{2} \frac{\left(\frac{\partial f}{\partial \alpha_i}(\alpha) - \frac{\partial f}{\partial \alpha_n}(\alpha) \right)^2}{K_{ii} - 2K_{in} + K_{nn}} \;\middle|\; 1 \le n \le \ell \text{ with } \alpha_n > L_n \right\} \ , \tag{3}$$

where the function to be maximized is a simple upper bound of the gain which is easier to compute than $g_B(\alpha)$. This very efficient algorithm is used in the popular software LIBSVM [3]. Both methods are known to converge to the optimum [2, 7, 3].

Another approach presented by Glasmachers and Igel [4] is to directly maximize the gain. Due to efficiency we can not maximize over all possible pairs. Instead we restrict ourselves to working sets such that one index is chosen from the working set selected in the most recent iteration. Let $\tilde{B}$ denote the most recent working set, then the Maximum Gain (MG) policy selects

$$B = (i, j) = \arg\max \left\{ g_B(\alpha) \;\middle|\; B \text{ is related to } \tilde{B} \right\} \ .$$

We call two tuples (i_1, j_1) and (i_2, j_2) related if the corresponding sets $\{i_1, j_1\}$ and $\{i_2, j_2\}$ have a common element. To select related pairs in subsequent iterations exactly captures the idea to reselect one previously selected variable.

However, to ensure convergence MG has to fall back to another algorithm whenever the most recent step ended near a corner, that is, both variables indexed by the previous working set end up in a small neighborhood of their lower or upper bounds. Originally, MVP selection was used in this case, but the second order selection (3) as well as any other convergent algorithm would be alternative options. The resulting algorithm is called Hybrid Maximum Gain (HMG). The handling of different cases and the reselection of one index from the previous working set considerably complicate the convergence proof given in [4]. Nevertheless the reselection of one index is worth the efforts because it significantly speeds up the whole algorithm for large scale problems where only small fractions of the kernel matrix fit into the cache. This is because exactly the rows of K corresponding to the indices in the working set are needed in each iteration. By reselecting one index it is ensured that the corresponding row was used recently and is available from the cache. In this paper we consider algorithms exploiting this feature.

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

3 Main Result

In this section we derive a general result for cache friendly SMO variants that reselect one element of the previous working set.

It is well known that the sequence produced by the SMO algorithm strictly increases the objective function value if and only if the working set selection algorithm selects a violating pair in each iteration. This condition is necessary but not sufficient for convergence to the optimum. For $B = (i, j)$ we define the vector $v_{(i,j)} = v_B = e_i - e_j$, where e_n is the n-th unit vector in $\mathbb{R}^\ell$. Further, we call a variable α_n free if it is not at the box boundary, that is, $L_n < \alpha_n < U_n$.

Definition 1. A working set $B = (i, j)$ is a violating pair for α if $\alpha_i < U_i$, $\alpha_j > L_j$ and $v_B^T \nabla f(\alpha) > 0$.

It is well known that, for non-optimal α, all working set selection algorithms discussed in the previous section select violating pairs. In particular for any non-optimal feasible α there exists a violating pair. Note that for (i, j) with $\alpha_i < U_i$ and $\alpha_j > L_j$ the quantity $v_{(i,j)}^T \nabla f(\alpha)$ is positive if and only if (i, j) is a violating pair.

The importance of violating pairs for the convergence of the SMO algorithm to the optimum motivates the following simple but important lemma:

Lemma 2. Let α be a non-optimal feasible point and i an index such that the corresponding variable α_i is free. Then there exists j such that either $B = (i, j)$ or $B = (j, i)$ is a violating pair. This pair fulfills the inequality

$$v_B^T \nabla f(\alpha) \geq \frac{1}{2} \max \left\{ v_W^T \nabla f(\alpha) \;\middle|\; W \text{ is a violating pair for } \alpha \right\} .$$

Proof Let $W = (m, n)$ be the maximizer of the right hand side, that is, (m, n) is the most violating pair defined in equation (2). In particular we have $\alpha_m < U_m$ and $\alpha_n > L_n$. From $v_{(m,i)} + v_{(i,n)} = v_{(m,n)}$ we conclude $v_{(m,i)}^T \nabla f(\alpha) + v_{(i,n)}^T \nabla f(\alpha) = v_{(m,n)}^T \nabla f(\alpha) > 0$ which implies that at least one of the summands is positive. This already shows that one of the pairs (m, i) or (i, n) is violating. We define the value $M = \frac{1}{2} \left(\frac{\partial f}{\partial \alpha_m}(\alpha) + \frac{\partial f}{\partial \alpha_n}(\alpha) \right)$ and write

$$\frac{\partial f}{\partial \alpha_m}(\alpha) = \left(M + \frac{1}{2} v_{(m,n)}^T \nabla f(\alpha) \right) \quad \text{and} \quad \frac{\partial f}{\partial \alpha_n}(\alpha) = \left(M - \frac{1}{2} v_{(m,n)}^T \nabla f(\alpha) \right) .$$

Depending on the derivative $\frac{\partial f}{\partial \alpha_i}(\alpha)$ we distinguish two cases: For $\frac{\partial f}{\partial \alpha_i}(\alpha) \leq M$ the computation

$$v_{(m,i)}^T \nabla f(\alpha) = \frac{\partial f(\alpha)}{\partial \alpha_m} - \frac{\partial f(\alpha)}{\partial \alpha_i} \geq \left(M + \frac{1}{2} v_{(m,n)}^T \nabla f(\alpha) \right) - M = \frac{1}{2} v_{(m,n)}^T \nabla f(\alpha)$$

reveals the desired estimate for $v_{(m,i)}^T \nabla f(\alpha)$ which immediately implies that (m, i) is a violating pair. It is completely analogous to obtain $v_{(i,n)}^T \nabla f(\alpha) \geq \frac{1}{2} v_{(m,n)}^T \nabla f(\alpha)$ with violating pair (i, n) from $\frac{\partial f}{\partial \alpha_i}(\alpha) \geq M$. ∎

ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

The result of this lemma is quite general. In the following we will apply it to the cache-friendly version of the SMO algorithm selecting related pairs in subsequent iterations:

Corollary 3. Let (i, j) be the working set of the previous SMO step resulting in the current point α. If α_i or α_j is free then there exists a violating pair B related to (i, j).

Proof First we consider the case that α_i is free. Then Lemma 2 states that there exists n such that either (i, n) or (n, i) is a violating pair. Obviously these pairs are related to (i, j). The proof is completely analogous if α_j is the free variable. ∎

If we further assume that the algorithm selects the new index j like in the proof of Lemma 2 resulting in the lower bound for $v_{(i,j)}^T \nabla f(\alpha)$ we can apply a powerful general convergence result by Chen et al. [5]. We quote this result in our terms:

Theorem 4 [Theorem 3 from [5]]. Assume the SMO algorithm produces an infinite sequence $(\alpha^{(n)})_{n\in\mathbb{N}}$ and let there exist $0 < \sigma \leq 1$ such that in each non-optimal α the working set B fulfills

$$v_B(\alpha)^T \nabla f(\alpha) \geq \sigma \max \left\{ v_W(\alpha)^T \nabla f(\alpha) \;\middle|\; W \text{ is a violating pair for } \alpha \right\} .$$

Then the sequence $f(\alpha^{(n)})$ converges to the optimum of problem (1).

Lemma 2 states that we can always find a working set related to the previous working set with $\sigma = \frac{1}{2}$. Thus, it seems as if the above theorem can guarantee convergence when selecting subsequent related working sets. The only shortcoming of this approach is that at least one variable indexed by the previous working set must be free. This is why HMG uses a fall-back algorithm without the restriction to reselect one index in this situation. However, in practice this situation occurs only in the very early stage of the optimization and thus does not affect the overall runtime of the cache friendly SMO algorithm (see [4]).

4 Application to the HMG Algorithm

Finally, we will apply Lemma 2 and in particular Corollary 3 to simplify the proof of convergence of the HMG algorithm given in [4]. This proof crucially depends on the following lemma which is used several times therein:

Lemma 5 [Lemma 8 from [4]]. We consider problem (1), a current non-optimal point α, and a previous working set B_1. If at least one of the variables indexed by B_1 is free then there exists a working set B_2 related to B_1 such that positive gain $g_{B_2}(\alpha) > 0$ can be achieved.

This lemma is obviously equivalent to Corollary 3. However, its formulation has two decisive drawbacks. First, it does not really capture the nature of the problem it solves. This is accomplished by Lemma 2. Second, its original proof (see [4]) is unnecessarily complicated.

The main part of the proof considers the case of general sub-problems with four variables. These problems are categorized into a finite number of cases. A total of $9^4 \cdot 3^2 \cdot 2^4 = 2,834,352$ different cases occur and a computer program is provided that checks each case individually. This proceeding is of course completely correct, but it is neither easy to verify by the reader nor does it provide deeper insights. In addition, it lacks mathematical beauty.

In contrast, in each of the proofs of Lemma 2 and Corollary 3 there are only two different cases, and these are clearly separable (which is reflected by the separation into Lemma 2 and Corollary 3). Therefore we can now enjoy the advantage of a proof conforming to a typical mathematical line of thought. Lemma 5 is indeed the most critical part of the proof presented in [4]. Therefore it is of major importance for the whole convergence result that this lemma is funded on a rigorous proof which can be easily validated and understood.

5 Conclusion

We present a result for proving the convergence of cache-friendly variants of the SMO algorithm to the optimum. Because of its generality it may even turn out to be valuable in other circumstances. The point of view taken in this work greatly simplifies the earlier approach in [4] and yields an equivalent result. It identifies the core of the restriction imposed by selecting related working sets in subsequent iterations. This much cleaner proceeding results in a more general formulation and allows for a greatly simplified proof.

References

[1] J. Platt. Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 12, pages 185–208. MIT Press, 1999.

[2] S. S. Keerthi and E. G. Gilbert. Convergence of a generalized SMO algorithm for SVM classifier design. *Machine Learning*, 46:351–360, 2002.

[3] R.-E. Fan, P.-H. Chen, and C.-J. Lin. Working Set Selection using the Second Order Information for Training Support Vector Machines. *Journal of Machine Learning Research*, 6:1889–1918, 2005.

[4] T. Glasmachers and C. Igel. Maximum Gain Working Set Selection for SVMs. *Journal of Machine Learning Research*, 7:1437–1466, 2006.

[5] P.-H. Chen, R.-E. Fan, and C.-J. Lin. A Study on SMO-type Decomposition Methods for Support Vector Machines. *IEEE Transactions on Neural Networks*, 17:893–908, 2006.

[6] T. Joachims. Making large-Scale SVM Learning Practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, chapter 11, pages 169–184. MIT Press, 1999.

[7] N. Takahashi and T. Nishi. Rigorous Proof of Termination of SMO Algorithm for Support Vector Machines. *IEEE Transaction on Neural Networks*, 16(3):774–776, 2005.