AN EFFICIENT SMO-LIKE ALGORITHM FOR MULTICLASS SVM

Fabio Aiolli
Dipartimento di Informatica
Corso Italia 40 56100, Pisa, Italy
E-mail: aiolli@di.unipi.it

Alessandro Sperduti
Dipartimento di Matematica Pura ed Applicata
Via Belzoni, 7, 35131 Padova, Italy
E-mail: sperduti@math.unipd.it

Abstract. Starting from a reformulation of Cramer & Singer Multiclass Kernel Machine, we propose a Sequential Minimal Optimization (SMO) like algorithm for incremental and fast optimization of the lagrangian. The proposed formulation allowed us to define very effective new pattern selection strategies which lead to better empirical results.

INTRODUCTION

In a multiclass classification framework for each instance it is required to associate one of $m \geq 2$ labels. When $m = 2$ the problem is said to be binary. Let $S = \{(\bar{x}_1, y_1), ..., (\bar{x}_n, y_n)\}$, $\bar{x}_i \in \Re^d$, be a set of training examples, we assume that each label y_i belongs to a set of cardinality m, i.e. $y_i \in \{1, ..., m\}$. We search for a multiclass classifier, i.e. a function $H(\bar{x})$ that maps an instance $\bar{x}$ to its correct label. Multiclass learning algorithms have been defined either by extending existing binary algorithms (see for example [3, 5]) or by devising new specialized algorithms (see for example [1]).

Kernel based machines as SVM [6] represent state of the art binary classification models with an easy and clear formulation of the related problem. In [2] an original formulation specialized for the multiclass case has been proposed. The classifier derived by this formulation is of the form

$$H(x) = \arg\max_{r \in R} \bar{W}_r \cdot \bar{x},$$

where W is a matrix of size $m \times d$, $\bar{W}_r$ is the rth row of M and R is the set of prototype indexes r each one associated to exactly one class.

Starting from the model in [2], we give a simplified formulation of the Multiclass Kernel-based Machine that explicitly shows the strict relationship

between this method and the original binary SVM. We also give an SMO-like optimization algorithm and new strategies for the selection of the examples to optimize that try to overcome some problems we observed with the original approach.

THE MULTICLASS CLASSIFICATION TASK

In [2] the authors propose a convincing formulation for the multiclass classification setting whose aim is to search for a set of vectors of small norm in a way such that the margin of classification, i.e. the difference between the positive score and the largest of the negative ones, is greater than 1. Formally:

$$(\bar{W}_{y_i} - \bar{W}_r) \cdot \bar{x} \geq 1, \ \forall i, r \neq y_i.$$

Furthermore, to allow for margin violations, they add soft margin slack variables $\xi_i \geq 0$, one for each example, and propose to encode all the constraints in a compact quadratic formulation:

$$\begin{cases} \min_{W,\bar{\xi}} \frac{1}{2}\beta||W||^2 + \sum_i \xi_i \\ s.t. \ \forall i, r, \ \bar{W}_{y_i} \cdot \bar{x}_i + \delta^r_{y_i} - \bar{W}_r \cdot \bar{x}_i \geq 1 - \xi_i \end{cases} \tag{1}$$

where $\delta^r_{y_i} \in \{0, 1\}$ is equal to 1 iif $r = y_i$. In this case the constraints will reduce to the positivity requirements for the slack variables ξ_i.

After a derivation that we omit here, they obtain a new equivalent formulation for the problem:

$$\begin{cases} \max_{\bar{\tau}} Q(\bar{\tau}) = \beta \sum_i \bar{\tau}_i \cdot \bar{1}_{y_i} - \frac{1}{2} \sum_{i,j} (\bar{\tau}_i \cdot \bar{\tau}_j) K(\bar{x}_i, \bar{x}_j) \\ s.t. \ \forall i, \ \bar{\tau}_i \leq \bar{1}_{y_i} \text{ and } \bar{\tau}_i \cdot \bar{1} = 0 \end{cases} \tag{2}$$

where we denoted $\bar{1}_y$ the vector whose components are all 0 except for the y-th component that is equal to 1 and $\bar{0}$ (resp. $\bar{1}$) denotes the vector whose components are all 0's (resp. 1's).

The authors then decompose this problem into multiple optimization problems of reduced size. In particular, they propose to isolate the contribution of the variables associated with a single example. In this way they can obtain n problems of reduced size, each one defined over m variables and with $m + 1$ constraints:

$$\begin{cases} \min_{\bar{\tau}} Q(\bar{\tau}) = \frac{1}{2} A_p (\bar{\tau}_p \cdot \bar{\tau}_p) + \bar{B}_p \cdot \bar{\tau}_p \\ s.t. \ \bar{\tau}_p \leq \bar{1}_{y_p} \text{ and } \bar{\tau}_p \cdot \bar{1} = 0 \end{cases} \tag{3}$$

where A_p and $\bar{B}_p$ are constants defined for each example p.

Then, they derive a fixed-point based algorithm able to efficiently find an approximated solution for the above reduced problem. The optimal solution for the whole problem is obtained by iterating on different examples. The efficiency of this scheme is tightly linked to the strategy based on which the examples are selected for optimization. By using the KKT conditions of the

problem (2) the authors derive a quantity $\psi_i \geq 0$ for each example and show that in the optimum this value needs to be equal to zero. Then, they use this value to drive the optimization process. In the baseline implementation the example that maximizes ψ_i is selected. In summary, their algorithm may be understood as a main loop which is composed of an example selection, via the ψ_i quantity, an invocation of a fixed-point algorithm that is able to approximate the solution of the associated problem as defined in (3) and the computation of the new value of ψ_i for each example. On each iteration, most computation time is spent on the last step since it requires the computation of one row of the kernel matrix, that one relative to the pattern with respect to which we have just optimized. This is why it is so important a strategy that tries to minimize the total number of patterns selected for optimization. In the same work, this point is attacked by maintaining an active set containing the subset of patterns having $\psi_i \geq \epsilon$ where ϵ is a suited accuracy threshold. Cooling schemes, i.e. heuristics based on the gradual decrement of this accuracy parameter, are used for improving the efficiency with large datasets.

In our opinion, this approach has some drawbacks:

i) the baseline method is not guaranteed to converge since it is possible that variables associated to the example that most violate the KKT conditions for the problem (2) may be optimal for the isolated problem (3) and thus it is not possible to further improve the lagrangian;

ii) cooling schemes indirectly solve the previous problem, however they do not always perform well;

iii) on each iteration, the fixed point optimization algorithm is executed from scratch, and previously computed solutions obtained for an example can't help when the same example is chosen again in future iterations; In addition it finds just an *approximated* solution for the associated pattern-related problem;

In the next sections we propose an incremental solution that is based on a variant of the well known SMO algorithm. This new method is equally simple as the original, however it is more robust and seems to be faster.

A NEW FORMULATION

In this section we restate the problem (2) in a such way we are able to obtain a simpler analysis of the algorithms proposed in this paper. With this new formulation in mind, it will be highlighted the strict relationships between this model and the original binary version of SVM's.

Let us simply set $C = \beta^{-1}$ and to introduce a new variable for each lagrangian coefficient

$$\alpha_i^r = C y_i^r \tau_i^r$$

where we denoted y_i^r to be equal to 1 if $r = y_i$ and -1 otherwise. Using the new variables, the constraints can be rewritten as

$$\alpha_i^r \geq 0 \text{ and } \alpha_i^{y_i} \leq C \text{ and } \sum_r y_i^r \alpha_i^r = 0 \Rightarrow \alpha_i^r \geq 0 \text{ and } \alpha_i^{y_i} = \sum_{r \neq y_i} \alpha_i^r \leq C$$

Now, by using the fact that $\alpha_i^{y_i} = \frac{1}{2} \sum_r \alpha_i^r$ and setting $s_r(\bar{x}_i) = \sum_j y_j^r \alpha_j^r K(\bar{x}_i, \bar{x}_j)$, the score for $\bar{x}$ with respect to the class r, after a few algebra we obtain

$$\begin{array}{rcl} Q(\bar{\alpha}) & = & \frac{1}{2C^2} \left(\sum_{r,i} \alpha_i^r - \sum_{r,i,j} y_i^r y_j^r \alpha_i^r \alpha_j^r K(\bar{x}_i, \bar{x}_j) \right) \\ & = & \frac{1}{2C^2} \sum_{r,i} \alpha_i^r \left(1 - y_i^r s_r(\bar{x}_i)\right) \end{array} \tag{4}$$

Finally, dropping constants that do not influence the solution of the problem, we can restate the original problem as:

$$\begin{cases} \max_{\bar{\alpha}} Q(\bar{\alpha}) = \sum_{r,i} \alpha_i^r (1 - y_i^r s_r(\bar{x}_i)) \\ s.t.\ \forall i, r,\ \alpha_i^r \geq 0 \text{ and } \alpha_i^{y_i} = \sum_{r \neq y_i} \alpha_i^r \leq C \end{cases} \tag{5}$$

and the decision function will be $H(\bar{x}) = arg\max_{r=1}^{m} s_r(\bar{x})$.

The formulation given in (5) strongly recalls the original formulation of SVM when the bias term is not applied and it results equivalent when using only two prototypes in a binary task. In fact, in this case, the second constraint asserts that, for each pattern, the α's for the two prototypes have the same value. Dropping multiplicative constants and considering $\alpha_i' = 2\alpha_i$ ad $C' = 2C$, we obtain the common formulation of the binary SVM

$$\begin{cases} \max_{\bar{\alpha}'} Q(\bar{\alpha}') = \sum_i \bar{\alpha}_i' - \frac{1}{2} \sum_{i,j} y_i^r y_j^r \bar{\alpha}_i' \bar{\alpha}_j' K(\bar{x}_i, \bar{x}_j) \\ s.t.\ \forall i,\ 0 \leq \bar{\alpha}_i' \leq C' \end{cases} \tag{6}$$

AN SMO-LIKE ALGORITHM

In this section we present an algorithm that is able to efficiently optimize the problem in (5). It is based on the SMO algorithm [4] for SVM and consists in updating two parameters at the time while keeping the solution in the feasible set. Like SMO, at each step, we choose to solve the smallest possible optimization problem which, in our case, still involves only two variables. Moreover, since the linear constraint $\sum_r y_p^r \alpha_p^r = 0$ is defined over the prototypes, the two variables involved must be associated to the same example.

We can now show how it is possible, given a pattern p and two prototype indexes r_1 and r_2, $r_1 \neq r_2$, to analytically solve the associated minimal problem. It is simple to show that, in order for the new α's to be still in the feasible set, i.e. $\sum_r y_p^r \alpha_p^r = 0$, given a value ν, the type of update must be restricted to one of the form:

$$\alpha_p^{r_1} \leftarrow \alpha_p^{r_1} + \epsilon_p^{r_1} \text{ and } \alpha_p^{r_2} \leftarrow \alpha_p^{r_2} + \epsilon_p^{r_2}, \text{ where } \epsilon_p^{r_1} = y_p^{r_1} \nu \text{ and } \epsilon_p^{r_2} = -y_p^{r_2} \nu.$$

Now, let us consider how the objective function changes with α. When perturbing the value of the α's, we have:

$$Q(\bar{\alpha}+\bar{\epsilon}) = \sum_{r,i}(\alpha_i^r+\epsilon_i^r) - \sum_{r,i} y_i^r(\alpha_i^r+\epsilon_i^r){s'}_r(x_i)$$

where we have denoted with ${s'}_r(\bar{x}_i)$ the score of the pattern $\bar{x}_i$ obtained with the updated $\bar{\alpha}$'s. When the update is made for a given pattern p, we have ${s'}_r(\bar{x}_i) = s_r(\bar{x}_i) + y_p^r\epsilon_p^r K_{ip}$ and

$$Q(\bar{\alpha}+\bar{\epsilon}) = Q(\bar{\alpha}) + \sum_r \epsilon_p^r - 2\sum_r y_p^r\epsilon_p^r s_r(\bar{x}_p) - \sum_r(\epsilon_p^r)^2 K_{pp}. \tag{7}$$

Moreover, by using the equations above, it is easy to derive that an update ν will change the value of the lagrangian of the following amount:

$$V_{r_1,r_2}^p(\nu) = 2\nu\left(\frac{y_p^{r_1}-y_p^{r_2}}{2} - s_{r_1}(\bar{x}_p) + s_{r_2}(\bar{x}_p) - \nu K_{pp}\right) \tag{8}$$

Since the above equation is concave on ν, the maximum gain for the value of the lagrangian (always greater or equal to 0) is obtained when the first derivative of $V(\nu)$ is equal to zero, that is for:

$$\nu = \frac{1}{2K_{pp}}\left(\frac{y_p^{r_1}-y_p^{r_2}}{2} - s_{r_1}(\bar{x}_p) + s_{r_2}(\bar{x}_p)\right) \tag{9}$$

Up to now, we have neglected the other two constraints $\alpha_p^r \geq 0$ and $\alpha_p^{y_p} = \sum_{r\neq y_p}\alpha_p^r \leq C$. For this constraints to be fulfilled the value of ν must be chosen such that

$$\alpha_p^{r_1} + y_p^{r_1}\nu \geq 0 \text{ and } \alpha_p^{r_2} - y_p^{r_2}\nu \geq 0 \tag{10}$$

and, if one of the two parameters chosen for optimization is the positive one:

$$\nu \leq 2\frac{C-\alpha_p^{y_p}}{y_p^{r_1}-y_p^{r_2}} \tag{11}$$

The above considerations are quite general and suggest a pratical algorithm to solve the problem in (5). This can be done by selecting pairs of variables and optimizing with respect to them until some stopping criterion is fulfilled. Moreover, equation (8) gives a natural method for the selection of the two variables involved, i.e. take the two indexes that maximize the value of (8). Finally, once chosen two variables to optimize, an analytic solution for this basic problem is given in (9). If this solution violates the constraints on α's, the ν value is scaled so to satisfy both eq.(10) and eq.(11). This algorithm will be referred to as Basic-SMO and it is depicted in Figure 1.

The same method can also be used as a basic step for an alternative method to the Cramer&Singer fixed-point algorithm for the optimization over a single example. In fact, by fixing an example and iterating multiple times the step described above on pairs of variables chosen among that associated to the pattern into consideration, it is guaranteed to find the optimality

BasicSMO(φ_V)

t=0;

repeat

$t \leftarrow t+1$

Heuristically choose an example p and two indexes $r_1 \neq r_2$

$\nu = \frac{\frac{1}{2}(y_p^{r_1} - y_p^{r_2}) - s_{r_1}(\bar{x}_p) + s_{r_2}(\bar{x}_p)}{2K_{pp}}$

if $(\alpha_p^{r_1} + y_p^{r_1}\nu < 0)$ **then** $\nu = -y_p^{r_1}\alpha_p^{r_1}$

if $(\alpha_p^{r_2} - y_p^{r_2}\nu < 0)$ **then** $\nu = y_p^{r_2}\alpha_p^{r_2}$

if $(\alpha_p^{y_p} + \frac{1}{2}(y_p^{r_1} - y_p^{r_2})\nu > C)$ **then** $\nu = 2\frac{C - \alpha_p^{y_p}}{y_p^{r_1} - y_p^{r_2}}$

$V(t) = 2\nu\left(\frac{1}{2}(y_p^{r_1} - y_p^{r_2}) - s_{r_1}(\bar{x}_p) + s_{r_2}(\bar{x}_p) - \nu K_{pp}\right)$

$\alpha_p^{r_1} = \alpha_p^{r_1} + y_p^{r_1}\nu;\ \alpha_p^{r_2} = \alpha_p^{r_2} + y_p^{r_2}\nu;$

until $V(t) \leq \varphi_V$

Figure 1: SMO-like basic optimization algorithm

condition. It is trivial to note that this algorithm requires a single step in the binary case. Moreover, the algorithm may be considered incremental in the sense that the solution previously found for a given pattern forms the initial condition when the pattern is selected again for optimization. In Figure 2 the pseudo-code of the proposed algorithm is presented. At each step, the algorithm applies the basic step to the $m(m-1)/2$ pairs of variables associated with the pattern which has been chosen for optimization until a certain condition on the value of the increment of the lagrangian is verified. It must be noted that for each iteration the scores for every pattern in the training set must be updated before to be used for the selection phase.

SELECTION CRITERIA AND COOLING SCHEMES

Following the previous considerations, it is not difficult to define a number of selection criteria which seem to be promising. We consider the following three procedures which return a value V_p that we use for deciding if a pattern has to be selected for optimization. Specifically:

i) Original KKT as defined in Cramer&Singer (here denoted KKT): in this case, the value corresponds to the ψ_p;

ii) Approximate Maximum Gain (here denoted AMG): in this case the value is computed as: $\max_{r_1 \neq r_2} V^p_{r_1,r_2}$ as defined in eq. (8). Notice that this

is a lower bound of the actual increment in the lagrangian obtained when the pattern p is selected for optimization;

iii) True Maximum Gain (here denoted BMG): in this case the value is computed using iteratively eq. (7) and it represents the actual increment in the lagrangian obtained when the pattern p is selected for optimization.

At the begin of each iteration, a threshold ρ is computed. For each example of the training set one of the above strategies is applied to it and the example will be selected for optimization if the value returned is greater than the threshold. The definition of the threshold ρ can be done either by a cooling scheme or in a data dependent way. In our case, we tried the logarithmic cooling scheme since it has shown the best results for the original approach. In addition, we propose two new schemes for the derivation of the value ρ: MAX where the threshold is computed as $\rho = \mu \cdot \max_p V_p$, $0 \leq \mu \leq 1$, and MEAN where the threshold is computed as $\rho = \text{mean}(V_p)$;

EXPERIMENTS

Experiments comparing the proposed approach versus C&S algorithm were conducted using a dataset consisting of 10705 digits randomly taken from the NIST-3 dataset. The training set consisted of 5000 randomly chosen digits.

The optimization algorithm has been chosen among: *i)* Cramer&Singer original fixed-point procedure (here denoted CS); *ii)* SMO-like procedure on the isolated problem (here denoted SMO); *iii)* Basic SMO algorithm (here denoted BAS). In the first experiments we used a cache of kernels of size 3000 that was able to contain all the kernels of the support vectors.

Figure 3 shows the effect of the application of the logarithmic scheme of cooling to the different selection/optimization strategies. It is possible to note that even if the proposed selection strategies largely improve the convergence rate, the optimal solution can not be reached. This shows how cooling schemes of the same family of that proposed in the C&S paper are not suited for the new proposed selection strategies. This is mostly due to the fact that the logarithmic function is very slow and the value returned by the strategies are soon less then the threshold. In particular the log function remains on a value of about 0.1 for many iterations. While this value is pretty good for the accuracy of the KKT solution it is not sufficient for the other selection schemes. In figure 4 different heuristics for the computation of the value ρ of the selection strategy of the SMO-like algorithm are compared. In this case the very simple heuristics MAX and MEAN reach similar performance which is moreover much better than the baseline C&S scheme. In figure 5, given the heuristic MEAN, different selection strategies are compared. In this case, the new strategies slightly outperform the one based on KKT conditions. Actually, as we see after, this slight improvement is due to the big size of the cache of kernels that makes the algorithm do not suffering of the

large amount of time spent in the computation of kernels that are not present in cache.

In order to reproduce conditions similar to the ones occurring when dealing with large datasets, the size of the cache has been reduced to 100 rows. As it is possible to see in figure 6-a a decrease in the performance is evident for each method, however, this decrease becomes more evident when KKT conditions are used as the pattern selection strategy. From the same figure we can see also a quite poor performance when the basic version of the SMO-like is used as a global optimization method. This demonstrates how much important it is to solve the overall problem for a pattern at the time. In fact, this leads to a decrease of the total number of pattern selected for optimization and consequently to a decrease of the number of kernel computations. This put also in evidence how much it is the amount of time spent with kernel computation with respect to the amount of time spent in the optimization. Figure 6-b clearly shows that the same argumentation can be applied to the recognition accuracy.

Finally, experimental comparisons of this new algorithms with respect to previous work have been done on a number of datasets from UCI and from USPS and MNIST digit recognition datasets, obtaining results that confirm the high performance obtained in terms of generalization with respect to state of the art methods as already described in [2].

CONCLUSIONS

We proposed an incremental and fast SMO-like optimization algorithm and new pattern selection strategies to solve the Cramer&Singer formulation of the Multiclass Kernel Machine. This algorithms lead to better empirical performance in terms of efficiency.

REFERENCES

[1] F. Aiolli and A. Sperduti, "A re-weighting strategy for improving margins," **Artificial Intelligence Journal**, vol. 137/1-2, pp. 197–216, 2001.

[2] K. Crammer and Y. Singer, "On the Algorithmic Implementation of Multiclass Kernel-based Machines," **Journal of Machine Learning Research**, vol. 2(Dec), pp. 265–292, 2001.

[3] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error correcting output codes," **Journal of Artificial Intelligence Research**, vol. 2, pp. 263–286, 1995.

[4] J. Platt, "Fast training of Support Vector Machines using sequential minimal optimization," **Advances in Kernel Methods - Support Vector Learning**, 1998.

[5] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," **Machine Learning**, vol. 37, no. 3, pp. 1–40, 1999.

[6] V. Vapnik, **Statistical Learning Theory**, Wiley, 1998.

OptimizeOnPattern(x_p,φ_V)

$t = 0$, $V(0) = 0$. $\forall r$, $\alpha'_r = \alpha^r_p$, $s_r = s_r(\bar{x_p})$

do

$t \leftarrow t + 1$, $V(t) = 0$

For each $r_1 \neq r_2$

$\nu = \frac{\frac{1}{2}(y_p^{r_1} - y_p^{r_2}) - s_{r_1} + s_{r_2}}{2K_{pp}}$

if $(\alpha'_{r_1} + y_p^{r_1}\nu < 0)$ **then** $\nu = -y_p^{r_1}\alpha'_{r_1}$

if $(\alpha'_{r_2} - y_p^{r_2}\nu < 0)$ **then** $\nu = y_p^{r_2}\alpha'_{r_2}$

if $(\alpha'_{y_p} + \frac{1}{2}(y_p^{r_1} - y_p^{r_2})\nu > C)$ **then** $\nu = 2\frac{C - \alpha'_{y_p}}{y_p^{r_1} - y_p^{r_2}}$

$V(t) = V(t) + 2\nu\left(\frac{1}{2}(y_p^{r_1} - y_p^{r_2}) - s_{r_1} + s_{r_2} - \nu K_{pp}\right)$

$\alpha'_{r_1} = \alpha'_{r_1} + y_p^{r_1}\nu$; $\alpha'_{r_2} = \alpha'_{r_2} + y_p^{r_2}\nu$;

$s_{r_1} = s_{r_1} + y_p^{r_1}\nu K_{pp}$; $s_{r_2} = s_{r_2} - y_p^{r_2}\nu K_{pp}$

until $(\frac{V(t) - V(t-1)}{V(t)} \leq \varphi_V)$

return $\{\bar{\alpha}'\}$

Figure 2: SMO-like algorithm for the optimization of the variables associated with a given pattern

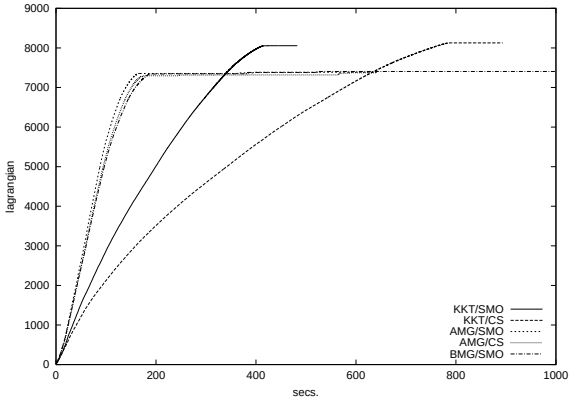


Figure 3: The effect of the logarithmic cooling scheme on different selection/optimization strategies.

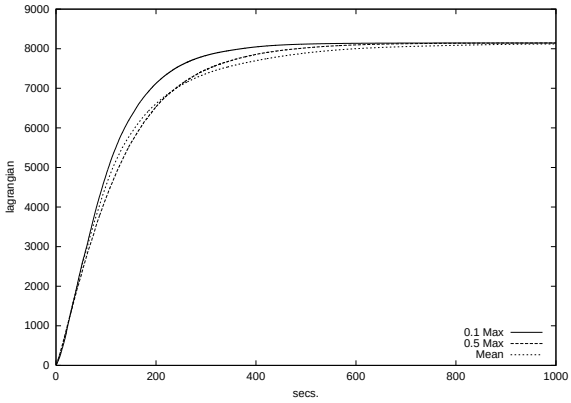


Figure 4: Comparison of different heuristics for the computation of the value ρ for the SMO-like algorithm.

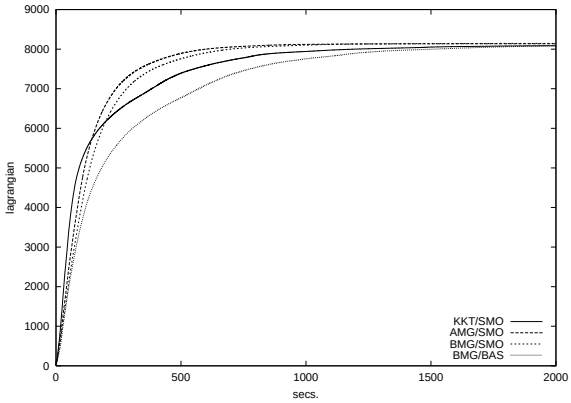
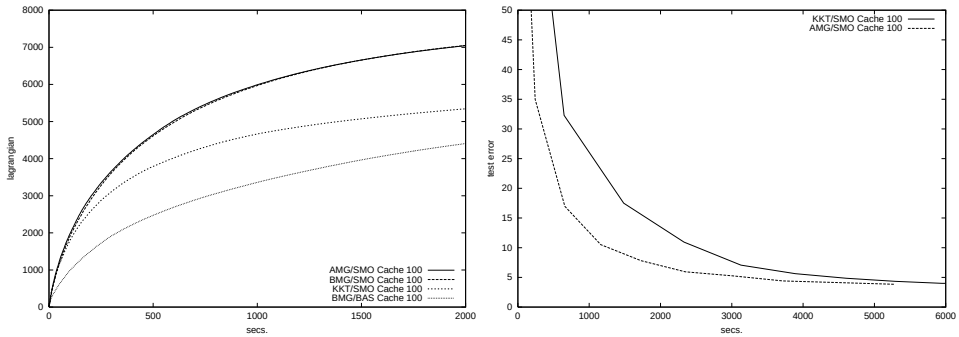


Figure 5: Comparison of different selection strategies using the heuristic MEAN.



(a) (b)

Figure 6: The effect of the cache limitation: (a) lagrangian value versus time; (b) test performance versus time.