Support Vector Machines: Heuristic of Alternatives (HoA)

Marcin Orchel
AGH Univeristy of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland

ABSTRACT

In this paper I will present a new heuristic for solving Support Vector Machines (SVM) optimization problem with Sequential Minimal Optimization (SMO) algorithm. SMO default heuristic chooses to the active set the two parameters based on SVM optimization conditions for two parameter subproblems. The proposed Heuristic of Alternatives (HoA) chooses parameters to the active set on the basis of not only SVM optimization conditions, but also on the basis of objective function value growth. Tests show that heuristic of alternatives is generally better than SMO default heuristic.

Keywords: Support Vector Machines, SVM, Sequential Minimal Optimization, SMO, Heuristic of Alternatives, HoA

1. SUPPORT VECTOR MACHINES

Support Vector Machine (SVM)\(^1\) is an approach to statistical classification. The idea of SVM is to separate classes with hyperplane by maximizing geometric distance between hyperplane and the nearest vectors.

In this paper Support Vector Machines will be used for binary classification. The classifier, which will be investigated in this article is the soft margin classifier with box constraints.\(^2\) This classifier is suited for many real world problems, because it can handle noisy data.

Learning SVM classifier leads to the following quadratic programming (QP) optimization problem:

**SVM optimization problem** \((O_1)\) Maximization of

\[
W(\vec{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K_{ij}
\]

with the constraints:

\[
\sum_{i=1}^{l} y_i \alpha_i = 0
\]

\[
0 \leq \alpha_i \leq C, \ i \in I = \{1, \ldots, l\}, \ C > 0
\]

Symbol meanings:

- \(\vec{\alpha}\) - parameters vector,
- \(l\) - size of the vector \(\vec{\alpha}\), number of data vectors,
- \(y_{ij}\) - abbreviation for \(y_i y_j\), \(y_i\) - classification value, \(y_i \in \{-1, 1\}\),
- \(C\) - soft margin classifier parameter,
- \(K_{ij}\) - abbreviation for \(K(\vec{x}_i, \vec{x}_j)\) - kernel function, where \(\vec{x}_i\) and \(\vec{x}_j\) are data vectors.

Further author information:
E-mail: marcin_orchel@o2.pl
1.1. Sequential Minimal Optimization

In order to speed up solving SVM optimization problem decomposition technique is used. Decomposition algorithm divides problem into smaller subproblems.

Because of complexity of algorithms solving $O_1$ for $l > 2$, the SVM problem is decomposed on two parameters subproblems, which are solved with analytical solution. The set of parameters chosen for subproblem is called active set. The algorithm which solves $O_1$ in this way is Sequential Minimal Optimization (SMO).

In every decomposition step optimization subproblem is solved for two chosen parameters:

**SMO optimization subproblem** ($O_2$) Maximization of

$$ W_2(\vec{\beta}) = \sum_{i=1}^{2} \beta_i + \sum_{i=1}^{l} \alpha_i - \sum_{j=1}^{2} y_{c_j} \beta_j \sum_{i=1}^{l} y_i \alpha_i K_{c_j i} - \frac{1}{2} \sum_{i=1}^{2} \beta_i^2 K_{c_i c_i} - \frac{1}{2} \sum_{i=1}^{l} y_{i} \alpha_i \sum_{j \in P, j \neq P} y_j \alpha_j K_{ij} $$

with the constraints:

$$ \sum_{i=1}^{2} y_{c_i} \beta_i + \sum_{i=1}^{l} y_i \alpha_i = 0 $$

$$ 0 \leq \beta_i \leq C, \ i \in \{1, 2\}, \ C > 0 $$

where

$P = \{c_1, c_2\}$ is a set of indices of parameters chosen to the active set, $c_1 \in I, c_2 \in I, c_1 \neq c_2$,

$\vec{\beta}$ is a subproblem variable vector,

$\beta_1$ is a searched value of $c_1$ parameter,

$\beta_2$ is a searched value of $c_2$ parameter.

The vector $\alpha$ is a previous solution. It must fulfill the constraints from $O_1$ problem.

2. SVM Heuristics for Two Parameters

SVM heuristic is responsible for choosing right parameters to the active set. In order to minimize computation time of learning SVM, heuristic should minimize the overall number of iterations. SVM heuristic for two parameters was proposed in. SVM default heuristic for two parameters proposed in this article is similar to existing one and is based on SVM optimization conditions for two parameter subproblems with linear constraint included.

We can transform linear constraint (2) to the form:

$$ \beta_1 = -y_{c_1} \beta_2 - y_{c_1} \sum_{i=1}^{l} y_i \alpha_i $$

After substituting $\beta_1$ to the (1) we get the following optimization subproblem:
SMO optimization subproblem with linear constraint included \((O_3)\)

Maximization of:

\[
W_3(\gamma) = -y_{c_1} \sum_{i \in P} y_i \alpha_i - y_{c_2} \gamma_1 + \gamma_1 + \sum_{i \in P} \alpha_i + \left( \sum_{i \in P} y_i \alpha_i \right) \left( \sum_{i \in P} y_i \alpha_i, K_{c_1} \right)
\]

\[
+ y_{c_2} \gamma_1 \sum_{i \in P} y_i \alpha_i (K_{c_1} - K_{c_1} - K_{c_2} - K_{c_2}) - \frac{1}{2} \left( \sum_{i \in P} y_i \alpha_i \right) K_{c_1} K_{c_2}
\]

\[- \frac{1}{2} \gamma_1^2 (K_{c_1} - 2K_{c_1} + K_{c_2} + K_{c_2}) - \frac{1}{2} \sum_{i \in P} \sum_{i \neq P} y_i \alpha_i \alpha_j K_{ij}
\]

with the constraints:

\[
0 \leq \gamma_1 \leq C, \ C > 0
\]

\[
0 \leq c = -y_{c_1} c_1 \gamma_1 - y_{c_2} \sum_{i \in P} y_i \alpha_i \leq C, \ C > 0
\]

where \(\gamma\) is a one element variable vector,
\(\gamma_1\) is a searched value of \(c_2\) parameter,
\(c\) is a searched value of \(c_1\) parameter.

The vector \(\alpha\) is a previous solution. It must fulfill the constraints from \(O_1\) problem.

The derivative of \(W_3(\gamma)\) in the point \(\gamma_1 = \alpha_{c_2}\) has a value:

\[
W'_3(\alpha_{c_2}) = y_{c_2} (E_1 - E_2)
\]

where

\[
E_1 = \sum_{i = 1}^l y_i \alpha_i K_{c_1} - y_{c_1}
\]

\[
E_2 = \sum_{i = 1}^l y_i \alpha_i K_{c_2} - y_{c_2}
\]

2.1. SVM default heuristic for two parameters

Necessary and sufficient SVM optimization conditions will be listed below. SVM default heuristic for choosing best parameters to the active set will be also presented.

Optimization conditions. Below we will analyze separately two cases: when \(\alpha_{c_2}\) is a bound parameter, and \(\alpha_{c_2}\) is a nonbound parameter.

1. \(\alpha_{c_2}\) is a bound parameter, \(\alpha_{c_2}\) is a bound or nonbound parameter. After changing parameter values, (4) must be fulfilled. It happens, when parameters \(c_1\) and \(c_2\) belong to the different groups \(G_1\) and \(G_2\) defined as:

\[
G_1 = \{ i : (y_i = 1 \land \alpha_i = 0) \lor (y_i = -1 \land \alpha_i = C) \lor (0 < \alpha_i < C) \}
\]

\[
G_2 = \{ i : (y_i = -1 \land \alpha_i = 0) \lor (y_i = 1 \land \alpha_i = C) \lor (0 < \alpha_i < C) \}
\]

Note that nonbound parameters are included in both groups.

Additionally it must be fulfilled:

\[
W'_3(\alpha_{c_2}) > 0 \text{ when } \alpha_{c_2} = 0
\]

\[
W'_3(\alpha_{c_2}) < 0 \text{ when } \alpha_{c_2} = C.
\]
After substitution (5) we get:
\[ y_{c_2}(E_1 - E_2) > 0 \text{ when } \alpha_{c_2} = 0 \]
\[ y_{c_2}(E_1 - E_2) < 0 \text{ when } \alpha_{c_2} = C \]

When \( y_{c_2} = 1 \):
\[ E_2 < E_1 \text{ when } \alpha_{c_2} = 0 \]
\[ E_2 > E_1 \text{ when } \alpha_{c_2} = C \]

When \( y_{c_2} = -1 \):
\[ E_2 > E_1 \text{ when } \alpha_{c_2} = 0 \]
\[ E_2 < E_1 \text{ when } \alpha_{c_2} = C \]

2. When \( \alpha_{c_2} \) is a nonbound parameter.

(a) If \( \alpha_{c_1} \) and \( \alpha_{c_2} \) are nonbound parameters optimization is possible when:
\[ W'_i(\alpha_{c_2}) \neq 0 \]

After substitution: \( E_1 \neq E_2 \).

(b) When \( \alpha_{c_1} \) is a bound parameter and \( \alpha_{c_2} \) is a nonbound parameter the optimization possibility can be computed by switching parameters. Thus we obtain the first case.

**Choosing the best pair.** Upon optimization conditions among pairs in which one of the parameters is bound (we assume it is \( c_2 \)), and both parameters belong to the opposite groups \( G_1 \) and \( G_2 \) the best pair for optimization will be that, for which \( m_i \) is maximal.

\( m_i \) is defined as:
- when parameter \( c_2 \) belongs to the \( G_1 \) group
  \[ m_i = E_1 - E_2 \]
- when parameter \( c_2 \) belongs to the \( G_2 \) group
  \[ m_i = E_2 - E_1 \]

The best pair in this case is a pair with maximal \( E \) from \( G_2 \) group and minimal \( E \) from \( G_1 \) group.

For pairs in which both parameters are nonbound and \( E_1 \neq E_2 \) we define:

\[ m_i = |E_1 - E_2| \]

The best pair to optimize in this case is that for which \( m_i \) is maximal.

After merging first and second case, the best pair to optimize will be with minimal \( E \) from \( G_1 \) group and with maximal \( E \) from \( G_2 \) group, if the chosen parameters are different.

After sorting parameters by \( E \) from maximal \( E \) in \( G_2 \) group and from minimal \( E \) in \( G_1 \) group we get the following lists:

\[ G_{1s} = (s_{11}, s_{12}, \ldots, s_{1p}) \]
\[ G_{2s} = (s_{21}, s_{22}, \ldots, s_{2q}) \]

where \( E_{sj} < E_{s_{j+1}} \) for \( j \in \{1, \ldots, p-1\} \) and
\( E_{s_{2j}} > E_{s_{2j+1}} \) for \( j \in \{1, \ldots, q-1\} \) and
\[ p + q \geq l. \]

The pair that will be chosen by SMO default heuristic is \((s_{11}, s_{21})\), if chosen parameters are different. If they are the same, the pair \((s_{12}, s_{21})\) or \((s_{11}, s_{22})\) will be compared and better pair for optimization will be used.

Comparison algorithm: if
\[ E_{s_{12}} - E_{s_{11}} > E_{s_{21}} - E_{s_{22}} \]
then \((s_{11}, s_{22})\) is a better pair, otherwise \((s_{12}, s_{21})\) is better.
3. HEURISTIC OF ALTERNATIVES (HOA)

SVM default heuristic choose parameters in every iteration based on SVM optimization conditions. The another measure of be closer to the solution is the objective function value of $O_1$ problem growth.

Heuristic of alternatives for the selected pairs of parameters compute objective function growth and choose the pair maximizing this growth. Both heuristic try to near to the solution the most in every iteration. Sometimes they choose the same parameters, sometimes not.

In heuristic of alternatives the strategy of generating pairs to check is to create pairs from parameters, which fulfill SVM optimization conditions the best or near the best. In the set of pairs there is always a pair, that would be chosen by SVM default heuristic. So the heuristic of alternatives has two strategies incorporated, one to check optimization conditions and the second to check objective function value growth.

The pairs that will be chosen for checking might look like this:

$$(s_{11}, s_{21}), (s_{12}, s_{21}), (s_{11}, s_{22}), (s_{13}, s_{21}), \ldots$$

The pair which has the maximal objective function value growth will be chosen.

In practice we use 4 the best pairs based on optimization conditions in alternatives set, or 9 pairs, or 16 pairs, for example:

$$(s_{11}, s_{21}), (s_{12}, s_{21}), (s_{11}, s_{22}), (s_{12}, s_{21})$$

In alternative sets we excluded pairs with both parameters the same.

3.1. Comparing time complexity of SMO default heuristic and heuristic of alternatives

SMO default heuristic time complexity

In every iteration optimization conditions must be computed. For every parameter we have to compute $E$ value. The complexity of computing $E$ value is $O(l)$. For all parameters and all iterations the complexity is $O(kl^2)$, where $k$ is a number of iterations.

Heuristic of alternatives time complexity

Objective function value growth of $O_1$ problem needs to be computed in every iteration for every alternative pair.

From the (1) we get the formula for objective function value growth:

$$\Delta W_2 (\vec{\beta}) = \sum_{i=1}^{2} \Delta \beta_i - \sum_{j=1}^{2} y_{c_j} \Delta \beta_j \sum_{i=1}^{l} y_i \alpha_i K_{c_j i} - \frac{1}{2} \sum_{i=1}^{2} \left( \beta_{i \text{new}}^2 - \beta_{i \text{old}}^2 \right) K_{c_1 c_i}$$

$$- y_{c_1} y_{c_2} \left( \beta_{1 \text{new}} \beta_{2 \text{new}} - \beta_{1 \text{old}} \beta_{2 \text{old}} \right) K_{c_1 c_2}$$

This step needs computing solution for all alternative pairs. Computing solution for single alternative pair has constant time. The complexity of computing objective function value growth for all iterations is $O(kml)$, where $m$ is a number of alternative pairs in every iteration. Overall complexity of heuristic of alternatives is $O(kl^2 + kml)$.

Both heuristics can be speed up by incorporating actualization of $E$ values for all parameters. After this modification computing optimization conditions for single parameter becomes constant. Complexity of SMO default heuristic falls to $O(kl)$. Computing objective function value growth also becomes constant for every parameter, so for the heuristic of alternatives the complexity is: $O(kl + km)$. The difference is the $km$ part, which doesn’t influence on overall time, when the number of parameters is big enough.
4. TESTING HEURISTIC OF ALTERNATIVES

Heuristic of alternatives will be compared with SMO default heuristic. The comparison will be on two levels:

- comparison of number of iterations,
- comparison of computing time.

**Data sets** Tests were done on the following data sets:

- data sets from images,
- stock exchange prediction data sets.

Data from images were extracted by getting the indices of every point and the color. Data vectors from images have two dimensions. Every point indices are the data vector coefficients and the classification is equal to 1, if the point color is closer to white, and is equal -1, if the point color is closer to black. Stock exchange prediction data sets were generated from end of day market data. Every vector corresponds to every market day and has two features. The first feature is a percentage close price growth from the day before previous day to the previous day. The second feature is a percentage volume growth from the day before previous day to the previous day. The classification is set to 1, when there were a close price growth from the previous day, and is set to -1, when there were a fall. The effects of slippage and order costs were omitted. This model is suited for trading during the day, buying in the beginning of the day, and selling in the end of the day.

**Data standarization** Data were squeezed proportionally to the [-1, 1] set, in order to minimize floating point representation errors. This operation were done independently for every feature as below:

\[ [a, b] \rightarrow [-1, 1] \]

The \(a\) and \(b\) were extracted from data set: \(a\) is a minimum value, \(b\) is the maximum value for the particular feature. When \(a \leq x \leq b\)

\(x\) value is changed to \(x'\) in the following way:

\[ x' = 2 \frac{x - a}{b - a} - 1. \]

**Test parameters**

Testing were done for various kernel functions:

- linear kernel
- polynomial kernel
- RBF kernel

Maximal number of alternatives in heuristic of alternatives is set to 4, 9, or 16.

The implementation includes the optimization described in the time complexity paragraph.

**Test 1** Data from images

Test parameters:

- 248 files
- every file has 108 vectors
<table>
<thead>
<tr>
<th>type of parameter</th>
<th>configuration 1</th>
<th>configuration 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heuristic</td>
<td>SMO default heuristic</td>
<td>Heuristic of Alternatives (HoA)</td>
</tr>
<tr>
<td><strong>Test .1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kernel</td>
<td>linear $a = 1$</td>
<td>the same</td>
</tr>
<tr>
<td><strong>Test .2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kernel</td>
<td>polynomial $a = 1.0$, dim = 2.0</td>
<td>the same</td>
</tr>
<tr>
<td><strong>Test .3</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kernel</td>
<td>RBF $\sigma = 1.0$</td>
<td>the same</td>
</tr>
</tbody>
</table>

Table 1. Test configurations

<table>
<thead>
<tr>
<th>Testing feature</th>
<th>SMO default</th>
<th>HoA (4)</th>
<th>SMO default</th>
<th>HoA (9)</th>
<th>SMO default</th>
<th>HoA (16)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test 1.1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of iterations</td>
<td>18172</td>
<td>15430</td>
<td>18172</td>
<td>13220</td>
<td>18172</td>
<td>11922</td>
</tr>
<tr>
<td>Number of tests with fewer iterations</td>
<td>23</td>
<td>219 (6)</td>
<td>4</td>
<td>243 (1)</td>
<td>2</td>
<td>246 (0)</td>
</tr>
<tr>
<td>Computation time</td>
<td>19.47</td>
<td>18.72</td>
<td>19.47</td>
<td>17.64</td>
<td>19.61</td>
<td>17.52</td>
</tr>
<tr>
<td>Number of tests with shorter times</td>
<td>94</td>
<td>112 (42)</td>
<td>69</td>
<td>148 (31)</td>
<td>60</td>
<td>148 (40)</td>
</tr>
<tr>
<td><strong>Test 1.2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of iterations</td>
<td>56734</td>
<td>39484</td>
<td>56734</td>
<td>32953</td>
<td>56734</td>
<td>28976</td>
</tr>
<tr>
<td>Number of tests with fewer iterations</td>
<td>21</td>
<td>210 (17)</td>
<td>7</td>
<td>239 (2)</td>
<td>4</td>
<td>242 (2)</td>
</tr>
<tr>
<td>Computation time</td>
<td>61.41</td>
<td>47.95</td>
<td>61.40</td>
<td>44.40</td>
<td>61.38</td>
<td>43.93</td>
</tr>
<tr>
<td>Number of tests with shorter times</td>
<td>57</td>
<td>172 (19)</td>
<td>56</td>
<td>176 (16)</td>
<td>53</td>
<td>182 (13)</td>
</tr>
<tr>
<td><strong>Test 1.3</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of iterations</td>
<td>21985</td>
<td>16986</td>
<td>21985</td>
<td>14047</td>
<td>21985</td>
<td>11934</td>
</tr>
<tr>
<td>Number of tests with fewer iterations</td>
<td>27</td>
<td>203 (18)</td>
<td>12</td>
<td>234 (2)</td>
<td>4</td>
<td>242 (2)</td>
</tr>
<tr>
<td>Computation time</td>
<td>25.13</td>
<td>22.21</td>
<td>25.30</td>
<td>19.67</td>
<td>25.41</td>
<td>18.03</td>
</tr>
<tr>
<td>Number of tests with shorter times</td>
<td>68</td>
<td>149 (31)</td>
<td>60</td>
<td>160 (28)</td>
<td>50</td>
<td>162 (36)</td>
</tr>
<tr>
<td><strong>All</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of iterations</td>
<td>96891</td>
<td>71900</td>
<td>96891</td>
<td>60220</td>
<td>96891</td>
<td>52832</td>
</tr>
<tr>
<td>Number of tests with fewer iterations</td>
<td>71</td>
<td>632 (41)</td>
<td>23</td>
<td>716 (5)</td>
<td>10</td>
<td>730 (4)</td>
</tr>
<tr>
<td>Computation time</td>
<td>106.01</td>
<td>88.88</td>
<td>106.16</td>
<td>81.71</td>
<td>106.40</td>
<td>79.47</td>
</tr>
<tr>
<td>Number of tests with shorter times</td>
<td>219</td>
<td>433 (92)</td>
<td>185</td>
<td>484 (75)</td>
<td>163</td>
<td>492 (89)</td>
</tr>
</tbody>
</table>

Table 2. Images data test results
\* \* C = 1.0 \*

Comparing configurations are in Tab. 1.

Heuristic comparison results are in the table Tab. 2.

**Test 2** Data from stock exchange prediction sets

Test parameters:

- 302 securities from Warsaw Stock Exchange
- every file has about 260 vectors, end of day data from August 2006 up to now
- \* C = 1.0 \*

Configurations are the same as in the image data tests.

Heuristic comparison results are in Tab. 3.

<table>
<thead>
<tr>
<th>Testing feature</th>
<th>SMO default</th>
<th>HoA (4)</th>
<th>SMO default</th>
<th>HoA (9)</th>
<th>SMO default</th>
<th>HoA (16)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test 2.1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of iterations</td>
<td>33817</td>
<td>28114</td>
<td>33817</td>
<td>26664</td>
<td>33817</td>
<td>26412</td>
</tr>
<tr>
<td>Number of tests with fewer iterations</td>
<td>30</td>
<td>259 (13)</td>
<td>12</td>
<td>285 (5)</td>
<td>13</td>
<td>284 (5)</td>
</tr>
<tr>
<td>Computation time</td>
<td>76.56</td>
<td>66.37</td>
<td>77.17</td>
<td>64.50</td>
<td>76.43</td>
<td>66.70</td>
</tr>
<tr>
<td>Number of tests with shorter times</td>
<td>54</td>
<td>200 (39)</td>
<td>50</td>
<td>223 (29)</td>
<td>66</td>
<td>206 (30)</td>
</tr>
<tr>
<td><strong>Test 2.2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of iterations</td>
<td>119725</td>
<td>80657</td>
<td>119725</td>
<td>65456</td>
<td>119725</td>
<td>59278</td>
</tr>
<tr>
<td>Number of tests with fewer iterations</td>
<td>15</td>
<td>287 (0)</td>
<td>7</td>
<td>295 (0)</td>
<td>4</td>
<td>297 (1)</td>
</tr>
<tr>
<td>Computation time</td>
<td>262.17</td>
<td>190.26</td>
<td>264.86</td>
<td>160.91</td>
<td>264.79</td>
<td>153.62</td>
</tr>
<tr>
<td>Number of tests with shorter times</td>
<td>26</td>
<td>265 (11)</td>
<td>19</td>
<td>273 (10)</td>
<td>20</td>
<td>272 (10)</td>
</tr>
<tr>
<td><strong>Test 2.3</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of iterations</td>
<td>67220</td>
<td>52436</td>
<td>67220</td>
<td>47155</td>
<td>67220</td>
<td>44568</td>
</tr>
<tr>
<td>Number of tests with fewer iterations</td>
<td>48</td>
<td>239 (15)</td>
<td>20</td>
<td>271 (11)</td>
<td>20</td>
<td>273 (9)</td>
</tr>
<tr>
<td>Computation time</td>
<td>159.32</td>
<td>128.40</td>
<td>159.17</td>
<td>120.74</td>
<td>159.39</td>
<td>119.10</td>
</tr>
<tr>
<td>Number of tests with shorter times</td>
<td>73</td>
<td>206 (23)</td>
<td>59</td>
<td>220 (23)</td>
<td>68</td>
<td>213 (21)</td>
</tr>
<tr>
<td><strong>All</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of iterations</td>
<td>220762</td>
<td>161207</td>
<td>220762</td>
<td>139275</td>
<td>220762</td>
<td>130258</td>
</tr>
<tr>
<td>Number of tests with fewer iterations</td>
<td>93</td>
<td>785 (28)</td>
<td>23</td>
<td>716 (5)</td>
<td>37</td>
<td>854 (15)</td>
</tr>
<tr>
<td>Computation time</td>
<td>498.05</td>
<td>385.04</td>
<td>501.20</td>
<td>346.14</td>
<td>500.60</td>
<td>339.41</td>
</tr>
<tr>
<td>Number of tests with shorter times</td>
<td>153</td>
<td>680 (73)</td>
<td>128</td>
<td>716 (62)</td>
<td>154</td>
<td>691 (61)</td>
</tr>
</tbody>
</table>

**Table 3.** Stock exchange data test results

**Conclusions** In the case of images data sets the heuristic of alternatives in the best version is faster than SMO default heuristic in more than 66% tests, and is slower in about 21%. Overall score is that heuristic of alternatives is faster than SMO default heuristic by 25%. In the case of stock exchange data sets the heuristic of alternatives is faster than SMO default heuristic in more than 76% tests, and is slower in about 16%. Overall score is that heuristic of alternatives is faster than SMO default heuristic by 32%.

**Time comparison of different heuristic of alternatives.** 9 alternatives version is better than 4 alternatives version for about 9%. A 16 alternatives version is faster than 9 alternatives version for about 2,3%.

Tests have shown, that heuristic of alternatives is generally better than SMO default heuristic.
Figure 1. Images data comparison of number of iterations of SMO default heuristic and heuristic of alternatives (16)

Figure 2. Images data comparison of computation times of SMO default heuristic and heuristic of alternatives (16)

Figure 3. Stocks data comparison of number of iterations of SMO default heuristic and heuristic of alternatives (16)
Figure 4. Stocks data comparison of computation times of SMO default heuristic and heuristic of alternatives (16)

5. ACKNOWLEDGMENTS

This research is founded by the Polish Ministry of Education and Science, Project No.3 T11F 010 30. I would like to express my sincere gratitude to Professor Witold Dzwinel (AGH University of Science and Technology, Institute of Computer Science) for contributing ideas, discussion and useful suggestions.

REFERENCES