Low rank updated LS-SVM classifiers for fast variable selection

Fabian Ojeda    Johan A.K. Suykens    Bart De Moor

Department of Electrical Engineering
Katholieke Universiteit Leuven
ESAT-SCD Division
Leuven, Belgium

May 23, 2008
Goals

- Select a subset of variables without degrade in prediction.
- Use of forward and backward search schemes.
- Exploit any (if possible) structure of the predictor.
- Reduce computational complexity.

Elements

- Objective function → easy/cheap to evaluate.
- Search procedure → efficient.
- Predictor → simple, yet powerful
Outline

Variable selection
  Methods

Approach

Least squares support vector machines (LS-SVM)

Leave-one out estimator
  Fast leave-one-out computation

Rank-one modifications
  Updating LS-SVM

Experiments
Variable selection

Definition
Given data $\mathcal{D} = \{(x_i, y_i)_{i=1}^n \mid x_i \in \mathbb{R}^d, y_i \in \{-1, +1\}\}$. Let the set of all variables $\mathcal{U} = \{u_1, \ldots, u_k, \ldots, u_d\}$.

Find a small subset $\mathcal{U}^* \subset \mathcal{U}$, $\mathcal{U}^* \in \mathbb{R}^p$, $p < d$, optimizing a primary objective function $J(\mathcal{U}^*)$, i.e. $J(\mathcal{U}^*) \leq J(\mathcal{U})$. 
Variable selection

Motivation

- *Effectiveness*: Reduce implementation costs.
- *Data understanding*: Identifying relevant factors.

Remark

- *NP-hard*: Combinatorial nature $2^d$.
- Multiple and different subsets may give same solution.
Methods

Univariate
One variable at a time.
- Statistical tests: fisher, $t$-test.
- Mutual information criteria.

Multivariate
- Filter techniques: independent of the predictor.
- Wrapper: Use predictor to evaluate subsets of variables and a search method, e.g. SFS, SBS.
- Embedded techniques: Variable selection embedded in the predictor, e.g. $\ell_1 - SVM$, $\ell_0 - SVM$. 
Approach

Elements

► Search method: forward and backward search.
► Objective function: leave-one-out error (LOO).
► Predictor/Classifier: least-squares support vector machines (LS-SVM).
► Speed up: rank-one matrix modifications.[Ojeda et al., 2008]
Least squares support vector machines (LS-SVM)

[Suykens et al., 2002]

**Primal formulation**

\[
\begin{align*}
\min_{w,b,e} & \quad \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^{n} e_i^2 \\
\text{s.t.} & \quad y_i = w^T \varphi(x_i) + b + e_i \\
& \quad i = 1, \ldots, n
\end{align*}
\]

Model: \( f(x) = w^T \varphi(x) + b \)

**Properties**

- Solution to a linear system. \( O(n^3) \).
- No sparseness.

**Dual formulation**

\[
\begin{bmatrix}
\Omega + \gamma^{-1} I_n & \mathbf{1} \\
\mathbf{1}^T & 0
\end{bmatrix}
\begin{bmatrix}
\alpha \\
b
\end{bmatrix} =
\begin{bmatrix}
y \\
0
\end{bmatrix}
\]

\( \Omega_{ij} = K(x_i, x_j) \)

\( f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x) + b \)
Alternative factorization

Positive definite system

\[ H = \Omega + \gamma^{-1}I_n \]

\[
\begin{bmatrix}
H & 0 \\
0 & 1^T H^{-1} 1
\end{bmatrix}
\begin{bmatrix}
\alpha + H^{-1} 1b \\
b
\end{bmatrix}
= \begin{bmatrix}
y \\
1^T H^{-1} y
\end{bmatrix},
\]

Model parameters

\[ b = 1^T H^{-1} y \left(1^T H^{-1} 1\right)^{-1} \]
\[ \alpha = H^{-1} (y - b 1) \]

Solution solely in terms of \( H^{-1} \).
Leave-one out (LOO) estimator

**Definition**

\[
\text{error} = \frac{1}{n} \sum_{i=1}^{n} \ell(f(-i)(x_i), y_i)
\]

- Unbiased estimator of the generalization error [Lachenbruch, 1967].
- Requires \(n\) runs of the learning algorithm.
- Non-smooth criterion.
- Alternatives: generalization error bounds [Chapelle et al., 2002].
Fast leave-one-out

Efficient leave-one-out → exploiting the structure of LS-SVM.

LS-SVM block structure

\[
\begin{bmatrix}
H \\
1^T \\
0
\end{bmatrix}
= \begin{bmatrix}
A_{11} & a_{12} \\
A_{12}^T & a_{22}
\end{bmatrix}
= A
\]

Closed-form

LOO residual for the \(i\)-th pattern is [Cawley, 2006]

\[
r_{i}^{(-i)} = y_{i} - \hat{y}_{i}^{(-i)} = \frac{\alpha_{i}}{(A^{-1})_{ii}}
\]

\[
\text{error} = \frac{1}{n} \sum_{i=1}^{n} \left(r_{i}^{(-i)}\right)^{2}
\]

\[s = -1^T H^{-1} 1, \quad \nu = H^{-1} 1\]

- Naive: \(n\) times LS – SVM → \(\mathcal{O}(n^4)\).
- Fast LOO: One LS – SVM → \(\mathcal{O}(n^3)\).
Rank-one updates/downdates

Update a given matrix $M$ once altered in some minimal sense.

Sherman-Morrison formula

Given $H^{-1} \in \mathbb{R}^{n \times n}$. Consider $M = H + URV^T$, where $U, V \in \mathbb{R}^{n \times q}$, $R \in \mathbb{R}^{q \times q}$, then

$$(H + URV^T)^{-1} = H^{-1} - H^{-1}UZ^{-1}V^TH^{-1}$$

where $Z = R^{-1} + V^TH^{-1}U$. Key: If $q \ll n$, then $R$ and $Z$ easier to invert than $M$ [Golub and Van Loan, 1989].

Symmetric rank-one update $q = 1$

If $q = 1$, i.e. $U = u \in \mathbb{R}^{n \times 1}$ and $R = 1$, the formula becomes

$$(H + uu^T)^{-1} = H^{-1} - \frac{H^{-1}uu^TH^{-1}}{1 + u^TH^{-1}u}$$
Updating LS-SVM solution

Compute LOO each time a variable is selected/removed. Linear kernels can be written in outer product form

$$\Omega = \begin{bmatrix} u_1, \ldots, u_d \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_d \end{bmatrix}^T = \sum_{k=1}^{d} u_k u_k^T$$

$$H = \Omega + \gamma^{-1}I_n = \sum_{k=1}^{d} u_k u_k^T + \gamma^{-1}I_n$$

At the level of variable \(k\) we have

$$H_k = \sum_{l=1}^{k-1} u_l u_l^T + \gamma^{-1}I_n + u_k u_k^T$$

$$H_k = H_{k-1} + u_k u_k^T$$

Avoid matrix inversions and use already available information.
Rank-one modifications

Updating LS-SVM

Updating LS-SVM solution

▶ Forward: Compute $H_k^{-1}$ at selection step $k$ from $H_{k-1}^{-1}$ at step $k - 1$, upon addition of variable $u_k$.

$$H_k^{-1} = H_{k-1}^{-1} - \frac{H_{k-1}^{-1} u_k u_k^T H_{k-1}^{-1}}{1 + u_k^T H_{k-1}^{-1} u_k}$$

No inverse matrix operations!!

▶ Backward: Compute $H_k^{-1}$ at removal step $k$ from $H_{k+1}^{-1}$ at step $k + 1$, upon removal of variable $u_k$

$$H_k^{-1} = H_{k+1}^{-1} + \frac{H_{k+1}^{-1} u_k u_k^T H_{k+1}^{-1}}{1 - u_k^T H_{k+1}^{-1} u_k}$$

Only the inverse of the matrix with all variables.
Experiments

Data

- Seven benchmark datasets.

Algorithms

- SVM-RFE with and without retraining [Guyon et al., 2002].
- Naive LS-SVM with forward selection.
- LS-SVM with fast LOO and rank-one modifications.

Validation

- Computational complexity.
- 10-fold cross-validation.
## Experiments

### Benchmark data

<table>
<thead>
<tr>
<th>Data</th>
<th>Training size</th>
<th>Test size</th>
<th>Realizations</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast cancer</td>
<td>200</td>
<td>77</td>
<td>100</td>
<td>9</td>
</tr>
<tr>
<td>Diabetes</td>
<td>468</td>
<td>300</td>
<td>100</td>
<td>8</td>
</tr>
<tr>
<td>Flare solar</td>
<td>666</td>
<td>400</td>
<td>100</td>
<td>9</td>
</tr>
<tr>
<td>German</td>
<td>700</td>
<td>300</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>Heart</td>
<td>170</td>
<td>100</td>
<td>100</td>
<td>13</td>
</tr>
<tr>
<td>Splice</td>
<td>1000</td>
<td>2175</td>
<td>20</td>
<td>60</td>
</tr>
<tr>
<td>Waveform</td>
<td>400</td>
<td>4600</td>
<td>100</td>
<td>21</td>
</tr>
</tbody>
</table>

### Microarray data

- Leukaemia. $n = 72, d = 7129$.
- Colon cancer $n = 60, d = 2000$. 
Experiments

Computational time in Colon data

Forward algorithms

Backward algorithms
Experiments

Classification performance

Benchmark data

- LS-SVM + rank-one update: Sequential forward selection (SFS).
- LS-SVM + rank-one downdate: Sequential backward elimination (SBE).
- Minimal number of variables between brackets []

<table>
<thead>
<tr>
<th>Data set</th>
<th>LS-SVM</th>
<th>LS-SVM+SFS</th>
<th>LS-SVM+SBE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast cancer</td>
<td>0.268 (0.044)</td>
<td>0.268 (0.052) [2/9]</td>
<td>0.264 (0.049) [2/9]</td>
</tr>
<tr>
<td>Diabetes</td>
<td>0.221 (0.013)</td>
<td>0.234 (0.017) [6/8]</td>
<td>0.234 (0.017) [6/8]</td>
</tr>
<tr>
<td>Flare solar</td>
<td>0.334 (0.014)</td>
<td>0.329 (0.018) [2/9]</td>
<td>0.326 (0.019) [2/9]</td>
</tr>
<tr>
<td>German</td>
<td>0.252 (0.020)</td>
<td>0.245 (0.021)</td>
<td>0.242 (0.022) [16/20]</td>
</tr>
<tr>
<td>Heart</td>
<td>0.151 (0.026)</td>
<td>0.157 (0.032)</td>
<td>0.157 (0.032)</td>
</tr>
<tr>
<td>Splice</td>
<td>0.161 (0.007)</td>
<td>0.161 (0.006) [27/60]</td>
<td>0.161 (0.007) [28/60]</td>
</tr>
<tr>
<td>Waveform</td>
<td>0.133 (0.006)</td>
<td>0.132 (0.006)</td>
<td>0.147 (0.009)</td>
</tr>
</tbody>
</table>
Experiments

Classification performance

Leukaemia data

![Graph showing test error vs number of ranked genes for Leukaemia data.]

Colon cancer data

![Graph showing test error in backward elimination for Colon cancer data.]

Test error in backward elimination. Alon data set.

- SVM-RFE1 – retraining
- SVM-RFE2 – no retraining
- LS-SVM+FastLO+RankDowndate
Use of leave-one-out as ranking criterion.

Rank-one modification into the linear kernel matrix to update inverse.

Applicable in high dimensional data.

Update, not recompute!
Selected references

Leave-one-out cross-validation based model selection criteria for weighted LS-SVMs.
In Proc of the International Joint Conference on Computational Neural Networks (IJCNN’06).

Choosing multiple parameters for support vector machines.

*Matrix computations.*

Gene selection for cancer classification using support vector machines.
*Machine learning*, 46.

Lachenbruch, P. (1967).
An almost unbiased method for the probability of misclassification in discriminant analysis.

Low rank updated LS-SVM classifiers for fast variable selection.

*Least Squares Support Vector Machines.*
World Scientific, Singapore.