LIBSVM 2.0: Solving Different Support Vector Formulations

Chih-Chung Chang and Chih-Jen Lin*

Abstract

LIBSVM (version 2.0) is an integrated tool for support vector classification and regression. It can also handle some variants of support vector machines (SVM): ν-SVM and one-class SVM. In this document, we briefly describe formulations implemented in LIBSVM.

1 Introduction

LIBSVM (version 2.0) is an integrated tool for support vector classification and regression. It can also handle some variants of support vector machines (SVM): ν-SVM and one-class SVM. In this document, we briefly describe formulations of these support vector machines.

A document of LIBSVM (version 1.0) which describes the main algorithm (for classification) is in (Chang and Lin 2000b).

The main difference between LIBSVM 1.0 and 2.0 is that the new version successfully implements classification, regression, and some variants of SVM in one short file (1000-line C++ code). Details of this implementation can be found in (Chang and Lin 2000a).

2 Formulations

2.1 C-Support Vector Classification

Given training vectors $x_i \in R^n, i = 1, \ldots, l$, in two classes, and a vector $y \in R^l$ such that $y_i \in \{1, -1\}$, C-SVC (Cortes and Vapnik 1995; Vapnik 1998) solves the following primal problem:

$$\begin{aligned} \min \quad & \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i \\ & y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \ldots, l. \end{aligned}$$

Its dual is

$$\begin{aligned} \min \quad & \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \\ & 0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, l, \\ & y^T\alpha = 0, \end{aligned} \tag{2.1}$$

*Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan (cjlin@csie.ntu.edu.tw).

where e is the vector of all ones, C is the upper bound, Q is an l by l positive semidefinite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, and $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is the kernel.

The decision function is

$$f(x) = sign(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b).$$

2.2 ν-Support Vector Classification

The ν-support vector classification (Schölkopf et al. 2000) uses a new parameter ν which let one control the number of support vectors and errors. The parameter $\nu \in (0, 1]$ is an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors.

Details of the algorithm implemented in LIBSVM can be found in (Chang and Lin 2000c). The primal form considered in (Chang and Lin 2000c) is:

$$\begin{aligned} &\min \frac{1}{2} w^T w + \frac{1}{2} b^2 - \nu\rho + \frac{1}{l} \sum_{i=1}^{l} \xi_i \\ &y_i (w^T \phi(x_i) + b) \geq \rho - \xi_i, \\ &\xi_i \geq 0, i = 1, \dots, l. \end{aligned}$$

The dual is:

$$\begin{aligned} &\min \frac{1}{2} \alpha^T (Q + yy^T) \alpha \\ &0 \leq \alpha_i \leq 1/l, \qquad i = 1, \dots, l, \\ &e^T \alpha = \nu, \end{aligned} \tag{2.2}$$

where $Q_{ij} \equiv y_i y_j K(x_i, x_j)$.

The decision function is:

$$f(x) = sign(\sum_{i=1}^{l} y_i \alpha_i (K(x_i, x) + 1)).$$

In LIBSVM, we solve a scaled version of (2.2):

$$\begin{aligned} &\min \frac{1}{2} \alpha^T (Q + yy^T) \alpha \\ &0 \leq \alpha_i \leq 1, \qquad i = 1, \dots, l, \\ &e^T \alpha = \nu l. \end{aligned}$$

We output α/ρ so the computed decision function is:

$$f(x) = sign(\sum_{i=1}^{l} y_i (\alpha_i / \rho)(K(x_i, x) + 1)).$$

By doing so, two marginal surfaces are

$$y_i(w^T\phi(x_i)+b)=\pm 1.$$

2.3 One-class Support Vector Machines

One-class SVM was proposed by Schölkopf et al. (1999) for estimating the support of a high-dimensional distribution. The primal form in (Schölkopf et al. 1999) is:

$$\begin{aligned}
&\min \frac{1}{2}w^T w-\rho+\frac{1}{\nu l}\sum_{i=1}^{l}\xi_i\\
&w^T\phi(x_i)\geq \rho-\xi_i,\\
&\xi_i\geq 0, i=1,\dots,l,\ \rho\geq 0.
\end{aligned}$$

Note that here we do not have class information any more. The dual is:

$$(D_\nu)\qquad \begin{aligned}
&\min \frac{1}{2}\alpha^T Q\alpha\\
&0\leq \alpha_i\leq 1/(\nu l), i=1,\dots,l,\\
&e^T\alpha=1,
\end{aligned}\tag{2.3}$$

where $Q_{ij}=K(x_i,x_j)\equiv \phi(x_i)^T\phi(x_j)$.

In LIBSVM we solve a scaled version of (2.3):

$$\begin{aligned}
&\min \frac{1}{2}\alpha^T Q\alpha\\
&0\leq \alpha_i\leq 1, \qquad i=1,\dots,l,\\
&e^T\alpha=\nu l.
\end{aligned}$$

The decision function is

$$f(x)=sign(\sum_{i=1}^{l}\alpha_i K(x_i,x)-\rho).$$

2.4 C-Support Vector Regression

The standard form of support vector regression (Vapnik 1998) is:

$$\begin{aligned}
&\min \frac{1}{2}w^T w+C\sum_{i=1}^{l}\xi_i+C\sum_{i=1}^{l}\xi_i^*\\
&z_i-w^T\phi(x_i)-b\leq \epsilon+\xi_i,\\
&w^T\phi(x_i)+b-z_i\leq \epsilon+\xi_i^*,\\
&\xi_i,\xi_i^*\geq 0, i=1,\dots,l.
\end{aligned}$$

The dual is:

$$\min \frac{1}{2}(\alpha-\alpha^*)^T Q(\alpha-\alpha^*) + \epsilon\sum_{i=1}^{l}(\alpha_i+\alpha_i^*) - \sum_{i=1}^{l} z_i(\alpha_i-\alpha_i^*)$$

$$\sum_{i=1}^{l}(\alpha_i-\alpha_i^*) = 0, 0 \le \alpha_i, \alpha_i^* \le C, i = 1, \ldots, l,$$

where $Q_{ij} = K(x_i, x_j) \equiv \phi(x_i)^T\phi(x_j)$.

The decision function is:

$$f(x) = \sum_{i=1}^{l}(\alpha_i-\alpha_i^*)K(x_i, x) + b.$$

3 Further Information

LIBSVM 2.0 is available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

Acknowledgments

This work was supported in part by the National Science Council of Taiwan via the grant NSC 89-2213-E-002-013. The authors thank Chih-Wei Hsu and Jen-Hao Lee for many helpful discussions and comments.

References

Chang, C.-C. and C.-J. Lin (2000a). An integrated implementation for support vector classification and regression. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan. In Preparation.

Chang, C.-C. and C.-J. Lin (2000b). LIBSVM: Introduction and benchmarks. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.

Chang, C.-C. and C.-J. Lin (2000c). Training ν-support vector classifiers: Theory and algorithms. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan. Submitted to Neural Computation.

Cortes, C. and V. Vapnik (1995). Support-vector network. *Machine Learning 20*, 273–297.

Schölkopf, B., J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson (1999). Estimating the support of a high-dimensional distribution. Technical Report 99-87, Microsoft Research.

Schölkopf, B., A. Smola, R. C. Williamson, and P. L. Bartlett (2000). New support vector algorithms. *Neural Computation 12*, 1083 – 1121.

Vapnik, V. (1998). *Statistical Learning Theory.* New York, NY: John Wiley.