# Outlier Path: A Homotopy Algorithm for Robust SVM

**Shinya SUZUMURA** SUZUMURA.MLLAB.NIT@GMAIL.COM
Nagoya Institute of Technology Gokiso-cho, Showa-ku, Nagoya, Aichi 466–8555 Japan

**Kohei OGAWA** OGAWA.MLLAB.NIT@GMAIL.COM
Nagoya Institute of Technology Gokiso-cho, Showa-ku, Nagoya, Aichi 466–8555 Japan

**Masashi Sugiyama** SUGI@CS.TITECH.AC.JP
Tokyo Institute of Technology O-okayama, Meguro-ku, Tokyo 152-8552, Japan

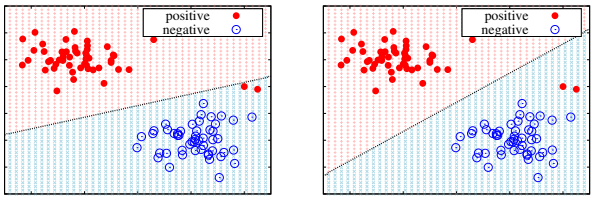
**Ichiro Takeuchi** TAKEUCHI.ICHIRO@NITECH.AC.JP
Nagoya Institute of Technology Gokiso-cho, Showa-ku, Nagoya, Aichi 466–8555 Japan

## Abstract

In recent applications with massive but less reliable data (e.g., labels obtained by a semi-supervised learning method or crowdsourcing), non-robustness of the support vector machine (SVM) often causes considerable performance deterioration. Although improving the robustness of SVM has been investigated for long time, robust SVM (RSVM) learning still poses two major challenges: obtaining a good (local) solution from a non-convex optimization problem and optimally controlling the robustness-efficiency trade-off. In this paper, we address these two issues simultaneously in an integrated way by introducing a novel *homotopy* approach to RSVM learning. Based on theoretical investigation of the geometry of RSVM solutions, we show that a path of local RSVM solutions can be computed efficiently when the influence of outliers is gradually suppressed as simulated annealing. We experimentally demonstrate that our algorithm tends to produce better local solutions than the alternative approach based on the concave-convex procedure, with the ability of stable and efficient model selection for controlling the influence of outliers.

## 1. Introduction

The *support vector machine* (SVM) is one of the most popular classification algorithms that has achieved significant empirical success in various real-world applications (Vapnik, 1996). However, SVM was known to be sensitive to outliers which limits the usability of SVM in recent applications with massive but less reliable data (e.g., automatically labeled data by semi-supervised learning or manually labeled data in crowdsourcing). In order to alleviate adverse influence of outliers, various robust extensions of SVM (robust SVM; RSVM) have been proposed (Masnadi-Shiraze & Vasconcelos, 2000; Shen et al., 2003; Krause & Singer, 2004; Liu et al., 2005; Liu & Shen, 2006; Xu et al., 2006; Collobert et al., 2006; Wu & Liu, 2007; Masnadi-Shirazi & Vasconcelos, 2009; Freund, 2009; Yu et al., 2010). Figure 1 illustrates the robust behavior of RSVM.




(a) Standard SVM (b) Robust SVM (RSVM)

*Figure 1.* Illustrative examples of (a) standard SVM and (b) robust SVM (RSVM) on a toy dataset. In RSVM, the classification result is not sensitive to the two red outliers in the right-hand side of the graphs.

When we use RSVM in practice, we encounter two major difficulties. The first one is the *non-convexity* of the RSVM optimization problem, which results in obtaining only a local optimal solution. Another difficulty is the control of the robustness of the solution. In RSVM, the ro-

*Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, 2014. JMLR: W&CP volume 32. 

bustness of the solution is controlled by a hyper-parameter, and we usually change the hyper-parameter value gradually and find the best one by cross-validation. However, due to the non-convexity, the RSVM solutions with slightly different hyper-parameter values can be significantly different, which makes model selection by cross-validation highly challenging.

In this paper, we introduce a novel approach to RSVM learning to address these issues. Our basic idea is to use the *homotopy* methods (Allgower & George, 1993; Gal, 1995; Ritter, 1984; Best, 1996) to trace a path of local optimal solutions when the influence of outliers is gradually decreased by changing the hinge loss to more robust ones. Figure 2 illustrates two different ways to gradually robustify the hinge loss. So far, homotopy-like methods have been (often implicitly) employed in sparse modeling and semi-supervised learning (Zhang, 2010; Mazumder et al., 2011; Zhou et al., 2012; Ogawa et al., 2013). However, to the best of our knowledge, this is the first work that applies the homotopy method to RSVM.

After problem formulation in § 2, we derive in § 3 the *necessary* and *sufficient* conditions for an RSVM solution to be locally optimal, and show that there exist a finite number of discontinuous points in the local solution path. We then propose an efficient algorithm in § 4 that can precisely detect such discontinuous points and *jump* to find a strictly better local optimal solution. In § 5, we experimentally demonstrate that our proposed method, named the *outlier path*, outperforms the existing RSVM algorithm based on the *concave-convex procedure* or the *difference-of-convex programming* (Shen et al., 2003; Krause & Singer, 2004; Liu et al., 2005; Liu & Shen, 2006; Collobert et al., 2006; Wu & Liu, 2007). Finally, we conclude in § 6.

## 2. Parameterized RSVM

Let us consider a binary classification problem with $n$ instances and $d$ features. We denote the training set as $\{(\boldsymbol{x}_i, y_i)\}_{i\in\mathbb{N}_n}$ where $\boldsymbol{x}_i \in \mathcal{X}$ is the input vector in the input space $\mathcal{X} \subset \mathbb{R}^d$, $y_i \in \{-1, 1\}$ is the binary class label, and the notation $\mathbb{N}_n := \{1, \ldots, n\}$ represents the set of natural numbers up to $n$. We write the decision function as $f(\boldsymbol{x}) := \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x})$, where $\boldsymbol{\phi}$ is the feature map implicitly defined by a kernel $\boldsymbol{K}$, $\boldsymbol{w}$ is a vector in the feature space, and $^\top$ denotes the transpose of vectors and matrices.

We introduce the following class of optimization problems *parameterized* by $\theta$ and $s$:

$$\min_{\boldsymbol{w}} \ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \ell(y_i f(\boldsymbol{x}_i);\ \theta, s), \tag{1}$$

where $C > 0$ is the regularization parameter. The loss function $\ell$ is characterized by a pair of parameters $\theta \in [0, 1]$ and $s \le 0$ in the following way:

$$\ell(z;\ \theta, s) := \begin{cases} [0, 1-z]_+, & z \ge s, \\ 1 - \theta z - s, & z < s, \end{cases} \tag{2}$$

where $[z]_+ := \max\{0, z\}$. We refer to $\theta$ and $s$ as *homotopy parameters*. Figure 2 shows the loss functions for several $\theta$ and $s$. The first homotopy parameter $\theta$ can be interpreted as the *weight* for an outlier: $\theta = 1$ indicates that the influences of outliers and inliers are same, while $\theta = 0$ indicates that outliers are completely ignored. The second homotopy parameter $s \le 0$ is interpreted as the threshold for outliers.

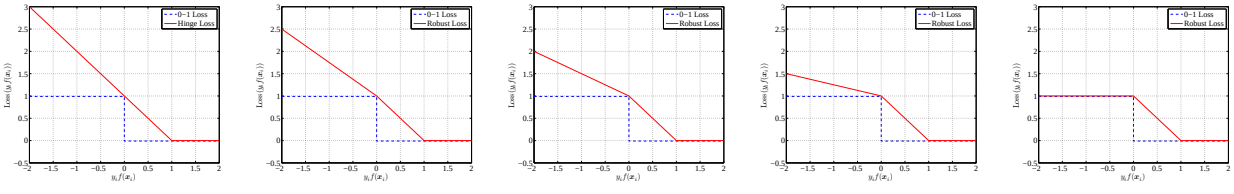
In the following sections, we consider two types of homotopy methods. In the first method, we fix $s = 0$, and gradually change $\theta$ from 1 to 0 (see the top five plots in Figure 2). In the second method, we fix $\theta = 0$ and gradually change $s$ from $-\infty$ to 0 (see the bottom five plots in Figure 2). Note that the loss function is reduced to the hinge loss for the standard (convex) SVM when $\theta = 1$ or $s = -\infty$. Therefore, each of the above two homotopy methods can be interpreted as the process of tracing a sequence of solutions when the optimization problem is gradually modified from convex to non-convex. We expect to find good local optimal solutions because such a process can be interpreted as *simulated annealing* (Hromkovic, 2001). In addition, we can adaptively control the degree of robustness by selecting the best $\theta$ or $s$ based on some model selection scheme.

## 3. Local Optimality of RSVM

In order to use the homotopy approach, we need to clarify the continuity of the local solution path. To this end, we investigate several properties of RSVM local solutions, and derive the necessary and sufficient conditions. Interestingly, our analysis reveals that the local solution path has a finite number of *discontinuous* points. The theoretical results presented here form the basis of our novel homotopy algorithm given in the next section that can properly handle the above discontinuity issue.

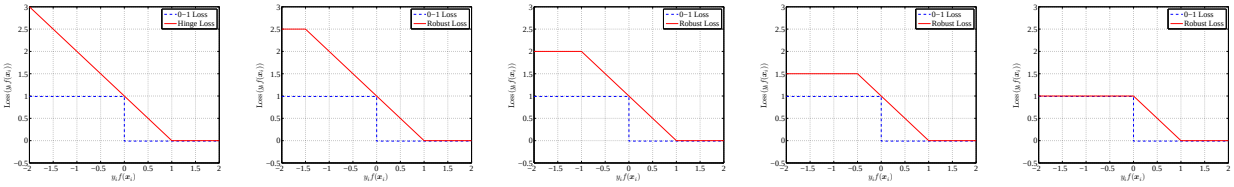
### 3.1. Conditionally Optimal Solutions

The basic idea of our theoretical analysis is to reformulate the RSVM learning problem as a combinatorial optimization problem. We consider a partition of the instances $\mathbb{N}_n := \{1, \ldots, n\}$ into two disjoint sets $\mathcal{I}$ and $\mathcal{O}$. The instances in $\mathcal{I}$ and $\mathcal{O}$ are defined as $\mathcal{I}$nliers and $\mathcal{O}$utliers, respectively. Here, we restrict that the margin $y_i f(\boldsymbol{x}_i)$ of an inlier should be larger than $s$, while that of an outlier should be smaller than $s$. We denote the partition as $\mathcal{P} := \{\mathcal{I}, \mathcal{O}\} \in 2^{\mathbb{N}_n}$, where $2^{\mathbb{N}_n}$ is the power set of $\mathbb{N}_n$. Given a partition $\mathcal{P}$, the above restrictions define the feasible re-



$\theta = 1, s = 0$ $\theta = 0.75, s = 0$ $\theta = 0.5, s = 0$ $\theta = 0.25, s = 0$ $\theta = 0, s = 0$

(a) Homotopy computation with decreasing $\theta$ from 1 to 0.



$\theta = 0, s = -\infty$ $\theta = 0, s = -1.5$ $\theta = 0, s = -1$ $\theta = 0, s = -0.5$ $\theta = 0, s = 0$

(b) Homotopy computation with decreasing $s$ from $-\infty$ to 0.

*Figure 2.* Robust loss functions for various homotopy parameters $\theta$ and $s$.

gion of the solution $f$ in the form of a convex polytope[1]:

$$\mathrm{pol}(\mathcal{P}; s) := \left\{ f \;\middle|\; \begin{array}{ll} y_i f(\boldsymbol{x}_i) \geq s, & i \in \mathcal{I} \\ y_i f(\boldsymbol{x}_i) \leq s, & i \in \mathcal{O} \end{array} \right\}. \tag{3}$$

Using the notion of the convex polytopes, the optimization problem (1) can be rewritten as

$$\min_{\mathcal{P} \in 2^{\mathbb{N}_n}} \left( \min_{f \in \mathrm{pol}(\mathcal{P}; s)} J_{\mathcal{P}}(f; \theta) \right), \tag{4}$$

where the objective function $J_{\mathcal{P}}$ is defined as[2]

$$J_{\mathcal{P}}(f; \theta) := \frac{1}{2} \|\boldsymbol{w}\|_2^2 + C \left( \sum_{i \in \mathcal{I}} [1 - y_i f(\boldsymbol{x}_i)]_+ + \theta \sum_{i \in \mathcal{O}} [1 - y_i f(\boldsymbol{x}_i)]_+ \right).$$

When the partition $\mathcal{P}$ is fixed, it is easy to confirm that the inner minimization problem in (4) is convex.

**Definition 1 (Conditionally optimal solutions)** *Given a partition $\mathcal{P}$, the solution of the following convex problem is said to be the conditionally optimal solution:*

$$f^*_{\mathcal{P}} := \operatorname*{argmin}_{f \in \mathrm{pol}(\mathcal{P}; s)} J_{\mathcal{P}}(f; \theta). \tag{5}$$

The formulation in (4) is interpreted as a combinatorial optimization problem of finding the best solution from all the $2^n$ conditionally optimal solutions $f^*_{\mathcal{P}}$ corresponding to all possible $2^n$ partitions[3].

Using the representer theorem or convex optimization theory, we can show that any conditionally optimal solution can be written as

$$f^*_{\mathcal{P}}(\boldsymbol{x}) := \sum_{j \in \mathbb{N}_n} \alpha^*_j y_j K(\boldsymbol{x}, \boldsymbol{x}_j), \tag{6}$$

where $\{\alpha^*_j\}_{j \in \mathbb{N}_n}$ are the optimal Lagrange multipliers. The following lemma summarizes the KKT optimality conditions of the conditionally optimal solution $f^*_{\mathcal{P}}$.

**Lemma 2** *The KKT conditions of the convex problem (5) is written as*

$$\begin{aligned}
y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) > 1 &\;\Rightarrow\; \alpha^*_i = 0, && \text{(7a)} \\
y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) = 1 &\;\Rightarrow\; \alpha^*_i \in [0, C], && \text{(7b)} \\
s < y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) < 1 &\;\Rightarrow\; \alpha^*_i = C, && \text{(7c)} \\
y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) = s, i \in \mathcal{I} &\;\Rightarrow\; \alpha^*_i \geq C, && \text{(7d)} \\
y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) = s, i \in \mathcal{O} &\;\Rightarrow\; \alpha^*_i \leq C\theta, && \text{(7e)} \\
y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) < s &\;\Rightarrow\; \alpha^*_i = C\theta. && \text{(7f)}
\end{aligned}$$

The proof is omitted because it can be easily derived based on standard convex optimization theory (Boyd & Vandenberghe, 2004).

### 3.2. The necessary and sufficient conditions for local optimality

From the definition of conditionally optimal solutions, it is clear that a local optimal solution must be conditionally

[1] Note that an instance with the margin $y_i f(\boldsymbol{x}_i) = s$ can be the member of either $\mathcal{I}$ or $\mathcal{O}$.

[2] Note that we omitted the constant terms irrelevant to the optimization problem.

[3] For some partitions $\mathcal{P}$, the convex problem (5) might not have any feasible solutions.

optimal within the convex polytope $\mathrm{pol}(\mathcal{P}; s)$. However, the conditional optimality does not necessarily indicate the local optimality as the following theorem suggests.

**Theorem 3** *For any $\theta \in [0, 1)$ and $s \leq 0$, consider a situation where a conditionally optimal solution $f^*_{\mathcal{P}}$ is at the boundary of the convex polytope $\mathrm{pol}(\mathcal{P}; s)$, i.e., there exists at least an instance such that $y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) = s$. In this situation, if we define a new partition $\tilde{\mathcal{P}} := \{\tilde{\mathcal{I}}, \tilde{\mathcal{O}}\}$ as*

$$\tilde{\mathcal{I}} \leftarrow \mathcal{I} \setminus \{i \in \mathcal{I} | y_i f^*(\boldsymbol{x}_i) = s\} \cup \{i \in \mathcal{O} | y_i f^*(\boldsymbol{x}_i) = s\}, \tag{8a}$$

$$\tilde{\mathcal{O}} \leftarrow \mathcal{O} \setminus \{i \in \mathcal{O} | y_i f^*(\boldsymbol{x}_i) = s\} \cup \{i \in \mathcal{I} | y_i f^*(\boldsymbol{x}_i) = s\}, \tag{8b}$$

*then the new conditionally optimal solution $f^*_{\tilde{\mathcal{P}}}$ is strictly better than the original conditionally optimal solution $f^*_{\mathcal{P}}$, i.e.,*

$$J_{\tilde{\mathcal{P}}}(f^*_{\tilde{\mathcal{P}}}; \theta) < J_{\mathcal{P}}(f^*_{\mathcal{P}}; \theta). \tag{9}$$

The proof is presented in Appendix A. Theorem 3 indicates that if $f^*_{\mathcal{P}}$ is at the boundary of the convex polytope $\mathrm{pol}(\mathcal{P}; s)$, i.e., if there is one or more instances such that $y_i f^*_{\mathcal{P}}(\boldsymbol{x}_i) = s$, then $f^*_{\mathcal{P}}$ is NOT locally optimal because there is a strictly better solution in the opposite side of the boundary.

The following theorem summarizes the necessary and sufficient conditions for local optimality. Note that, in non-convex optimization problems, the KKT conditions are necessary but not sufficient in general.

**Theorem 4** *For $\theta \in [0, 1)$ and $s \leq 0$,*

$$y_i f^*(\boldsymbol{x}_i) > 1 \quad \Rightarrow \quad \alpha^*_i = 0, \tag{10a}$$

$$y_i f^*(\boldsymbol{x}_i) = 1 \quad \Rightarrow \quad \alpha^*_i \in [0, C], \tag{10b}$$

$$s < y_i f^*(\boldsymbol{x}_i) < 1 \quad \Rightarrow \quad \alpha^*_i = C, \tag{10c}$$

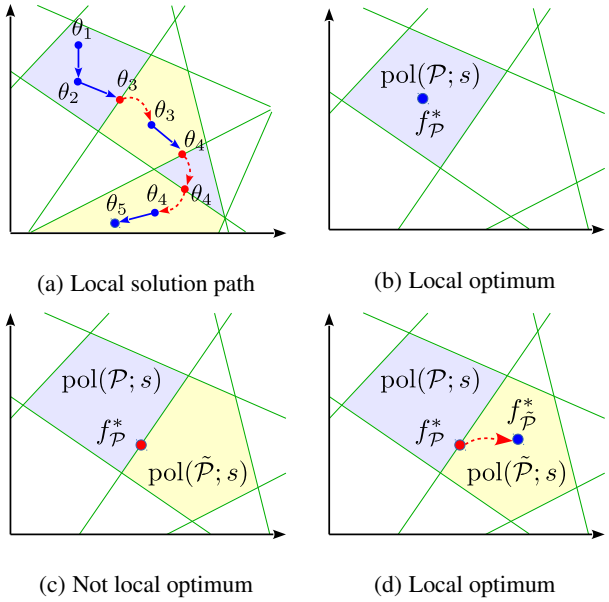
$$y_i f^*(\boldsymbol{x}_i) < s \quad \Rightarrow \quad \alpha^*_i = C\theta, \tag{10d}$$

$$y_i f^*(\boldsymbol{x}_i) \neq s, \quad \forall i \in \mathbb{N}_n, \tag{10e}$$

*are necessary and sufficient for $f^*$ to be locally optimal.*

The proof is presented in Appendix B. The condition (10e) indicates that the solution at the boundary of the convex polytope is not locally optimal. Figure 3 illustrates when a conditionally optimal solution can be locally optimal with a certain $\theta$ or $s$.

Theorem 4 suggests that, whenever the local solution path computed by the homotopy approach encounters a boundary of the current convex polytope at a certain $\theta$ or $s$, the solution is not anymore locally optimal. In such cases, we need to somehow find a new local optimal solution at that $\theta$ or $s$, and restart the local solution path from the new one. In other words, the local solution path has *discontinuity* at that $\theta$ or $s$. Fortunately, Theorem 3 tells us how to handle such a situation. If the local solution path arrives at the boundary, it can *jump* to the new conditionally optimal solution $f^*_{\tilde{\mathcal{P}}}$ which is located on the opposite side of the boundary. This jump operation is justified because the new solution is shown to be strictly better than the previous one. Figure 3 (c) and (d) illustrate such a situation.




(a) Local solution path (b) Local optimum

(c) Not local optimum (d) Local optimum

*Figure 3.* Solution space of RSVM. (a) The arrows indicate a local solution path when $\theta$ is gradually moved from $\theta_1$ to $\theta_5$ (see § 4 for more details). (b) $f^*_{\mathcal{P}}$ is locally optimal if it is at the strict interior of the convex polytope $\mathrm{pol}(\mathcal{P}; s)$. (c) If $f^*_{\mathcal{P}}$ exists at the boundary, then $f^*_{\mathcal{P}}$ is feasible, but not locally optimal. A new convex polytope $\mathrm{pol}(\tilde{\mathcal{P}}; s)$ defined in the opposite side of the boundary is shown in yellow. (d) A strictly better solution exists in $\mathrm{pol}(\tilde{\mathcal{P}}; s)$.

## 4. Outlier Path Algorithm

Based on the analysis presented in the previous section, we develop a novel homotopy algorithm for RSVM. We call the proposed method the *outlier-path (OP)* algorithm. For simplicity, we consider homotopy path computation involving either $\theta$ or $s$, and denote the former as OP-$\theta$ and the latter as OP-$s$. OP-$\theta$ computes the local solution path when $\theta$ is gradually decreased from 1 to 0 with fixed $s = 0$, while OP-$s$ computes the local solution path when $s$ is gradually increased from $-\infty$ to 0 with fixed $\theta = 0$.

### 4.1. Overview

The main flow of the OP algorithm is described in Algorithm 1. The solution $f$ is initialized by solving the standard (convex) SVM, and the partition $\mathcal{P} := \{\mathcal{I}, \mathcal{O}\}$ is defined to satisfy the constraints in (3). The algorithm mainly switches over the two steps called the *continuous step (C-*

**Algorithm 1** Outlier Path Algorithm

1: Initialize the solution $f$ by solving the standard SVM.
2: Initialize the partition $\mathcal{P} := \{\mathcal{I}, \mathcal{O}\}$ as follows:

$$\mathcal{I} \leftarrow \{i \in \mathbb{N}_n | y_i f(\boldsymbol{x}_i) \leq s\},$$
$$\mathcal{O} \leftarrow \{i \in \mathbb{N}_n | y_i f(\boldsymbol{x}_i) > s\}.$$

3: $\theta \leftarrow 1$ for OP-$\theta$; $s \leftarrow \min_{i \in \mathbb{N}_n} y_i f(\boldsymbol{x}_i)$ for OP-$s$.
4: **while** $\theta > 0$ for OP-$\theta$; $s < 0$ for OP-$s$ **do**
5: **if** $(y_i f(\boldsymbol{x}_i) \neq s \; \forall \; i \in \mathbb{N}_n)$ **then**
6: Run C-step.
7: **else**
8: Run D-step.
9: **end if**
10: **end while**

*step)* and the *discontinuous step (D-step)*.

In the C-step (Algorithm 2), a continuous path of local solutions is computed for a sequence of gradually decreasing $\theta$ (or increasing $s$) within the convex polytope $\text{pol}(\mathcal{P}; s)$ defined by the current partition $\mathcal{P}$. If the local solution path encounters a boundary of the convex polytope, i.e., if there exists at least an instance such that $y_i f(\boldsymbol{x}_i) = s$, then the algorithm stops updating $\theta$ (or $s$) and enters the D-step.

In the D-step (Algorithm 3), a better local solution is obtained for fixed $\theta$ (or $s$) by solving a convex problem defined over another convex polytope in the opposite side of the boundary (see Figure 3(d)). If the new solution is again at a boundary of the new polytope, the algorithm repeatedly calls the D-step until it finds the solution in the strict interior of the current polytope.

The C-step can be implemented by any homotopy algorithms for solving a sequence of quadratic problems (QP). In OP-$\theta$, the local solution path can be exactly computed because the path within a convex polytope can be represented as piecewise-linear functions of the homotopy parameter $\theta$. In OP-$s$, the C-step is trivial because the optimal solution is shown to be constant within a convex polytope. In § 4.2 and § 4.3, we will describe the details of our implementation of the C-step for OP-$\theta$ and OP-$s$, respectively.

In the D-step, we only need to solve a single quadratic problem (QP). Any QP solver can be used in this step. We note that the *warm-start* approach (DeCoste & Wagstaff, 2000) is quite helpful in the D-step because the difference between two conditionally optimal solutions in adjacent two convex polytopes is typically very small. In § 4.4, we describe the details of our implementation of the D-step. Figure 4 illustrates an example of the local solution path obtained by OP-$\theta$.

In Algorithm 1, If the conditionally optimal solution is at the boundary, we again enters to the D-step. The objective

**Algorithm 2** Continuous Step (C-step)

1: **while** $(y_i f(\boldsymbol{x}_i) \neq s \; \forall \; i \in \mathbb{N}_n)$ **do**
2: Solve the sequence of convex problems,

$$\min_{f \in \text{pol}(\mathcal{P}; s)} J_{\mathcal{P}}(f; \theta),$$

for gradually decreasing $\theta$ in OP-$\theta$ or gradually increasing $s$ in OP-$s$.
3: **end while**

**Algorithm 3** Discontinuous Step (D-step)

1: Update the partition $\mathcal{P} := \{\mathcal{I}, \mathcal{O}\}$ as follows:

$$\mathcal{I} \leftarrow \mathcal{I} \setminus \{i \in \mathcal{I} | y_i f(\boldsymbol{x}_i) = s\} \cup \{i \in \mathcal{O} | y_i f(\boldsymbol{x}_i) = s\},$$
$$\mathcal{O} \leftarrow \mathcal{O} \setminus \{i \in \mathcal{O} | y_i f(\boldsymbol{x}_i) = s\} \cup \{i \in \mathcal{I} | y_i f(\boldsymbol{x}_i) = s\}.$$

2: Solve the following convex problem for fixed $\theta$ and $s$:

$$\min_{f \in \text{pol}(\mathcal{P}; s)} J_{\mathcal{P}}(f; \theta).$$

function $J_{\mathcal{P}}$ strictly decreases each time as shown in Theorem 3. Since any local optimal solutions must be in the strict interior as shown in Theorem 4, and the number of convex polytopes is finite, the algorithm will finally find a local optimal solution in finite time.

### 4.2. Continuous-Step for OP-$\theta$

In the C-step, the partition $\mathcal{P} := \{\mathcal{I}, \mathcal{O}\}$ is fixed, and our task is to solve a sequence of convex quadratic problems (QPs) parameterized by $\theta$ within the convex polytope $\text{pol}(\mathcal{P}; s)$. It has been known in optimization literature that a certain class of parametric convex QP can be exactly solved by exploiting the piecewise linearity of the solution path (Best, 1996). We can easily show that the local solution path of OP-$\theta$ within a convex polytope is also represented as a piecewise-linear function of $\theta$. The algorithm presented here is similar to the *SVM regularization path* algorithm in Hastie et al. (2004).

Let us consider a partition of the inliers in $\mathcal{I}$ into the following three disjoint sets:

$$\mathcal{R} := \{i | 1 < y_i f(\boldsymbol{x}_i)\},$$
$$\mathcal{E} := \{i | y_i f(\boldsymbol{x}_i) = 1\},$$
$$\mathcal{L} := \{i | s < y_i f(\boldsymbol{x}_i) < 1\}.$$

For a given fixed partition $\{\mathcal{R}, \mathcal{E}, \mathcal{L}, \mathcal{O}\}$, the KKT conditions of the convex problem (5) indicate that

$$\alpha_i = 0 \; \forall \; i \in \mathcal{R}, \;\; \alpha_i = C \; \forall \; i \in \mathcal{L}, \;\; \alpha_i = C\theta \; \forall \; i \in \mathcal{O}.$$

The KKT conditions also imply that the remaining Lagrange multipliers $\{\alpha_i\}_{i \in \mathcal{E}}$ must satisfy the following linear

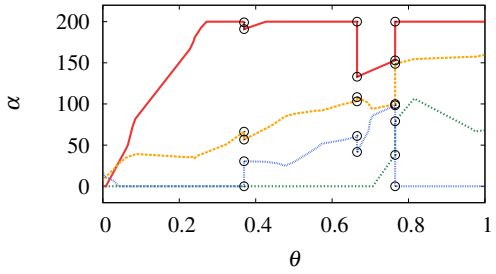



*Figure 4.* An example of the local solution path by OP-$\theta$ on a simple toy data set (with $C = 200$). The paths of five Lagrange multipliers $\alpha_1^*, \cdots, \alpha_4^*$ are plotted in the range of $\theta \in [0, 1]$. Open circles represent the discontinuous points in the path. In this simple example, we had experienced three discontinuous points at $\theta = 0.37, 0.67$ and $0.77$.

system of equations:

$$y_i f(\boldsymbol{x}_i) = \sum_{j \in \mathbb{N}_n} \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 \ \forall\, i \in \mathcal{E}$$
$$\Leftrightarrow \quad \boldsymbol{Q}_{\mathcal{E}\mathcal{E}} \boldsymbol{\alpha}_{\mathcal{E}} = \boldsymbol{1} - \boldsymbol{Q}_{\mathcal{E}\mathcal{L}} \boldsymbol{1} C - \boldsymbol{Q}_{\mathcal{E}\mathcal{O}} \boldsymbol{1} C \theta, \tag{11}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ is an $n \times n$ matrix whose $(i, j)^{\text{th}}$ entry is defined as $Q_{ij} := y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Here, a notation such as $\boldsymbol{Q}_{\mathcal{E}\mathcal{L}}$ represents a submatrix of $\boldsymbol{Q}$ having only the rows in the index set $\mathcal{E}$ and the columns in the index set $\mathcal{L}$. By solving the linear system of equations (11), the Lagrange multipliers $\alpha_i, i \in \mathbb{N}_n$, can be written as an affine function of $\theta$.

Noting that $y_i f(\boldsymbol{x}_i) = \sum_{j \in \mathbb{N}_n} \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is also represented as an affine function of $\theta$, any changes of the partition $\{\mathcal{R}, \mathcal{E}, \mathcal{L}\}$ can be exactly identified when the homotopy parameter $\theta$ is continuously decreased. Since the solution path linearly changes for each partition of $\{\mathcal{R}, \mathcal{E}, \mathcal{L}\}$, the entire path is represented as a continuous piecewise-linear function of the homotopy parameter $\theta$. We denote the points in $\theta \in [0, 1)$ at which members of the sets $\{\mathcal{R}, \mathcal{E}, \mathcal{L}\}$ change as *break-points* $\theta_{BP}$.

Using the piecewise-linearity of $y_i f(\boldsymbol{x}_i)$, we can also identify when we should switch to the D-step. Once we detect an instance satisfying $y_i f(\boldsymbol{x}_i) = s$, we exit the C-step and enter the D-step.

### 4.3. Continuous-Step for OP-$s$

Since $\theta$ is fixed to 0 in OP-$s$, the KKT conditions (7) yields

$$\alpha_i = 0 \ \forall\, i \in \mathcal{O}.$$

This means that outliers have no influence on the solution and thus the conditionally optimal solution $f_{\mathcal{P}}^*$ does not change with $s$ as long as the partition $\mathcal{P}$ is unchanged. The only task in the C-step for OP-$s$ is therefore to find the next $s$ that changes the partition $\mathcal{P}$. Such $s$ can be simply found as

$$s \ \leftarrow \ \min_{i \in \mathcal{L}} y_i f(\boldsymbol{x}_i).$$

### 4.4. Discontinuous-Step (for Both OP-$\theta$ and OP-$s$)

As mentioned before, any convex QP solver can be used for the D-step. When the algorithm enters the D-step, we have the conditionally optimal solution $f_{\mathcal{P}}^*$ for the partition $\mathcal{P} := \{\mathcal{I}, \mathcal{O}\}$. Our task here is to find another conditionally optimal solution $f_{\tilde{\mathcal{P}}}^*$ for $\tilde{\mathcal{P}} := \{\tilde{\mathcal{I}}, \tilde{\mathcal{O}}\}$ given by (8).

Given that the difference between the two solutions $f_{\mathcal{P}}^*$ and $f_{\tilde{\mathcal{P}}}^*$ is typically small, the D-step can be efficiently implemented by a technique used in the context of incremental learning (Cauwenberghs & Poggio, 2001).

Let us define

$$\Delta_{\mathcal{I} \to \mathcal{O}} := \{i \in \mathcal{I} \mid y_i f_{\mathcal{P}}(\boldsymbol{x}_i) = s\},$$
$$\Delta_{\mathcal{O} \to \mathcal{I}} := \{i \in \mathcal{O} \mid y_i f_{\mathcal{P}}(\boldsymbol{x}_i) = s\},$$

and $\boldsymbol{\alpha}^{(\text{bef})}$ be the corresponding $\boldsymbol{\alpha}$ at the beginning of the D-Step. Then, we consider the following parameterized problem with parameter $\mu \in [0, 1]$:

$$f_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i; \mu) := f_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i) + \mu \Delta f_i \ \forall\, i \in \mathbb{N}_n,$$

where

$$\Delta f_i := y_i \begin{bmatrix} \boldsymbol{K}_{i, \Delta_{\mathcal{I} \to \mathcal{O}}} & \boldsymbol{K}_{i, \Delta_{\mathcal{O} \to \mathcal{I}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_{\Delta_{\mathcal{I} \to \mathcal{O}}}^{(\text{bef})} - \boldsymbol{1} C \theta \\ \boldsymbol{\alpha}_{\Delta_{\mathcal{O} \to \mathcal{I}}}^{(\text{bef})} - \boldsymbol{1} C \end{bmatrix}.$$

We can show that $f_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i; \mu)$ is reduced to $f_{\mathcal{P}}(\boldsymbol{x}_i)$ when $\mu = 1$, while it is reduced to $f_{\tilde{\mathcal{P}}}(\boldsymbol{x}_i)$ when $\mu = 0$ for all $i \in \mathbb{N}_n$. By using a similar technique to incremental learning (Cauwenberghs & Poggio, 2001), we can efficiently compute the path of solutions when $\mu$ is continuously changed from 1 to 0. This algorithm behaves similarly to the C-step in OP-$\theta$. The implementation detail of the D-step is described in Appendix C.

## 5. Numerical Experiments

In this section, we compared the proposed outlier-path (OP) algorithm with the concave-convex procedure (CCCP) (Yuille & Rangarajan, 2002). In most of the existing RSVM studies, CCCP or a variant called difference of convex (DC) programming are used for optimizing RSVM (Shen et al., 2003; Krause & Singer, 2004; Liu et al., 2005; Liu & Shen, 2006; Collobert et al., 2006; Wu & Liu, 2007).

**Setup** We used the 10 benchmark data sets listed in Table 1. We randomly divided each data set into the training (40%), validation (30%), and test (30%) sets for training, model selection (including the selection of $\theta$ or $s$), and performance evaluation, respectively. In the training and validation data, we flipped 15% of the labels as outliers.

*Table 1.* Benchmark data sets. $n$ and $d$ denote the number of instances and the input dimensionality, respectively.

| Data | | $n$ | $d$ |
|---|---|---|---|
| D1 | BreastCancerDiagnostic | 569 | 30 |
| D2 | AustralianCreditApproval | 690 | 14 |
| D3 | German.Numer | 1000 | 24 |
| D4 | SVMGuide1 | 3089 | 4 |
| D5 | Spambase | 4601 | 57 |
| D6 | Musk | 6598 | 166 |
| D7 | Gisette | 6000 | 5000 |
| D8 | w5a | 9888 | 300 |
| D9 | a6a | 11220 | 122 |
| D10 | a7a | 16100 | 122 |

**Generalization Performance** First, we compared the generalization performance. We used the linear kernel and the radial basis function (RBF) kernel defined as $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right)$, where $\gamma$ is a kernel parameter fixed to $\gamma = 1/d$ with $d$ being the input dimensionality. Model selection was carried out by finding the best hyperparameter combination that minimizes the validation error. We have a pair of hyperparameters in each setup. In all the setups, the regularization parameter $C$ was chosen from $\{0.01, 0.1, 1, 10, 100\}$, while the candidates of the homotopy parameter $\theta$ or $s$ were set as follows:

- In OP-$\theta$, all the break-points $\theta_{BP}$ were considered as the candidates (note that the local solutions at each break-point have been already computed in the homotopy computation).
- In OP-$s$, all the break-points for $s_{BP}$ between $s_{\text{init}} := \min_{i\in\mathbb{N}_n} y_i f(\boldsymbol{x}_i)$ and 0 are considered as the candidates.
- In CCCP-$\theta$ (which is compared with OP-$\theta$), the homotopy parameter $\theta$ was selected from $\theta \in \{1, 0.75, 0.5, 0.25, 0\}$.
- In CCCP-$s$ (which is compared with OP-$s$), the homotopy parameter $s$ was selected from

$$s \in \{s_{\text{init}}, 0.75 s_{\text{init}}, 0.5 s_{\text{init}}, 0.25 s_{\text{init}}, 0\}.$$

Note that both OP and CCCP were initialized by using the standard SVM.

Tables 2 and 3 represent the average and the standard deviation of the test errors on 10 different random data splits. These results indicate that OP could find better local solutions and the degree of robustness was appropriately controlled.

**Computational Time** Finally, we compared the computational costs of the entire model-building process of each method. The results are shown in Figure 5. Note that the computational cost of the OP algorithm does not depend on the number of hyperparameter candidates of $\theta$ or $s$, because the entire path of local solutions has already been computed with the infinitesimal resolution in the homotopy computation. On the other hand, the computational cost of CCCP depends on the number of hyperparameter candidates. In our implementation of CCCP, we used the warm-start approach, i.e., we initialized CCCP with the previous solution for efficiently computing a sequence of solutions. The results indicate that the proposed OP algorithm enables stable and efficient control of robustness, while CCCP suffers a trade-off between model selection performance and computational costs.

*Table 2.* The mean of test error and standard deviation (linear). Smaller test error is better. The numbers in bold face indicate the better method in terms of the test error.

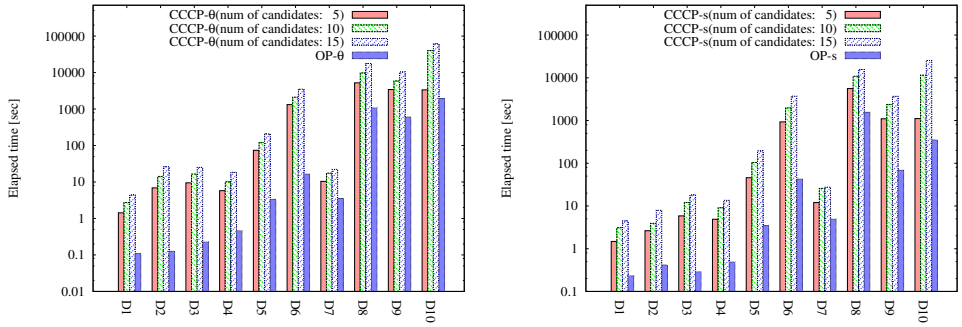
| Data | $C$-SVM | CCCP-$\theta$ | OP-$\theta$ | CCCP-$s$ | OP-$s$ |
|---|---|---|---|---|---|
| D1 | .056(.016) | .050(.014) | **.049(.016)** | .055(.018) | **.050(.016)** |
| D2 | .151(.018) | **.145(.007)** | .151(.018) | **.145(.007)** | .152(.010) |
| D3 | .281(.028) | .270(.033) | .270(.023) | **.262(.013)** | .266(.013) |
| D4 | .066(.007) | .047(.007) | .047(.005) | .053(.010) | **.042(.006)** |
| D5 | .108(.010) | .088(.009) | .088(.009) | .088(.010) | **.084(.007)** |
| D6 | .072(.005) | **.058(.006)** | .064(.003) | .061(.007) | **.060(.003)** |
| D7 | .185(.013) | .184(.010) | .184(.010) | .184(.010) | .184(.010) |
| D8 | .020(.002) | .020(.003) | .020(.002) | .021(.003) | **.020(.003)** |
| D9 | .173(.004) | .181(.009) | **.173(.005)** | .165(.004) | **.164(.004)** |
| D10 | .173(.008) | .176(.006) | **.173(.007)** | **.160(.004)** | .161(.005) |

*Table 3.* The mean of test error and standard deviation (RBF).

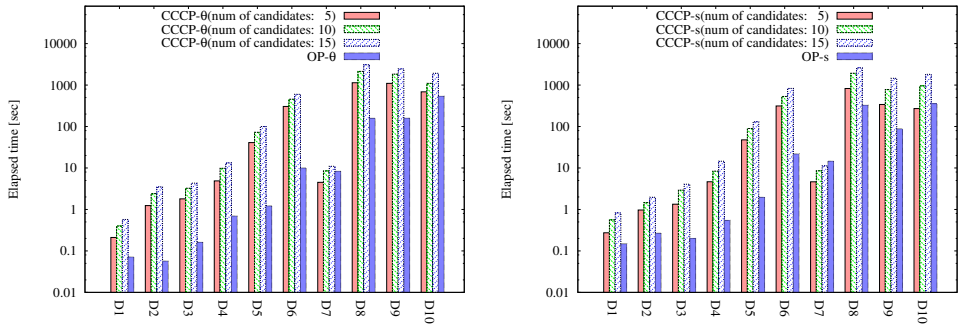
| Data | $C$-SVM | CCCP-$\theta$ | OP-$\theta$ | CCCP-$s$ | OP-$s$ |
|---|---|---|---|---|---|
| D1 | .055(.017) | .043(.022) | **.042(.017)** | **.037(.016)** | .038(.013) |
| D2 | .149(.010) | .148(.010) | **.147(.010)** | .146(.013) | **.142(.013)** |
| D3 | .276(.024) | .267(.026) | **.266(.024)** | .271(.015) | **.261(.020)** |
| D4 | .052(.009) | .048(.009) | **.044(.006)** | .047(.008) | **.040(.005)** |
| D5 | .117(.012) | .109(.013) | **.107(.012)** | .107(.011) | **.094(.008)** |
| D6 | .046(.007) | .045(.007) | .045(.007) | .045(.007) | **.043(.006)** |
| D7 | .044(.003) | .044(.003) | .044(.003) | .044(.003) | .044(.003) |
| D8 | .022(.003) | .022(.003) | .022(.003) | .022(.003) | **.021(.002)** |
| D9 | .169(.003) | .170(.005) | **.169(.004)** | .168(.005) | **.162(.003)** |
| D10 | .163(.003) | .163(.003) | .163(.003) | .162(.002) | **.160(.004)** |

## 6. Conclusions

In this paper, we proposed a novel robust SVM learning algorithm based on the homotopy approach that allows efficient computation of the sequence of local optimal solutions when the influence of outliers is gradually deemphasized. The algorithm is built on our theoretical findings about the geometric property and the optimality conditions of an RSVM local solution. Experimental results indicate that our algorithm tends to find better local solutions possibly due to the simulated annealing-like effect and the stable control of robustness. One of the important future works is to adopt scalable homotopy algorithms or approximate parametric programming algorithms (Giesen et al., 2012) as the building block of our algorithm to further improve the computational efficiency.




(a) Elapsed time for CCCP and proposed OP (linear)




(b) Elapsed time for CCCP and proposed OP (RBF)

*Figure 5.* Elapsed time when the number of $(\theta, s)$-candidates is increased. Changing the number of hyperparameter candidates affects the computation time of CCCP, but not OP because the entire path of solutions is computed with the infinitesimal resolution.

## Acknowledgments

The authors thank anonymous reviewers for their fruitful comments. MS was supported by JST CREST program. IT was also supported by JST CREST program, and MEXT Kakenhi 26280083 and 26106513.

## References


Allgower, E. L. and George, K. Continuation and path following. *Acta Numerica*, 2:1–63, 1993.

Best, M. J. An algorithm for the solution of the parametric quadratic programming problem. *Applied Mathemetics and Parallel Computing*, pp. 57–76, 1996.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.

Cauwenberghs, G. and Poggio, T. Incremental and decremental support vector machine learning. In *Advances in Neural Information Processing Systems*, volume 13, pp. 409–415. 2001.

Collobert, R., Sinz, F., Weston, J., and Bottou, L. Trading convexity for scalability. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 201–208, 2006.

DeCoste, D. and Wagstaff, K. Alpha seeding for support vector machines. In *Proceeding of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.

Freund, Y. A more robust boosting algorithm. *arXiv:0905.2138*, 2009.

Gal, T. *Postoptimal Analysis, Parametric Programming, and Related Topics*. Walter de Gruyter, 1995.

Giesen, J., Jaggi, M., and Laue, S. Approximating parameterized convex optimization problems. *ACM Transactions on Algorithms*, 9, 2012.

Hastie, T., Rosset, S., Tibshirani, R., and Zhu, J. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–415, 2004.

Hromkovic, J. *Algorithmics for Hard Problems*. Springer, 2001.

Krause, N. and Singer, Y. Leveraging the margin more carefully. In *Proceedings of the 21st International Conference on Machhine Learning*, pp. 63–70, 2004.

Liu, Y. and Shen, X. Multicategory $\psi$-learning. *Journal of the American Statistical Association*, 101:98, 2006.

Liu, Y., Shen, X., and Doss, H. Multicategory $\psi$-learning and support vector machine: Computational tools. *Journal of Computational and Graphical Statistics*, 14:219–236, 2005.

Masnadi-Shiraze, H. and Vasconcelos, N. Functional gradient techniques for combining hypotheses. In *Advances in Large Margin Classifiers*, pp. 221–246. MIT Press, 2000.

Masnadi-Shirazi, H. and Vasconcelos, N. On the design of loss functions for classification: theory, robustness to outliers, and savageboost. In *Advances in Neural Information Processing Systems*, volume 22, pp. 1049–1056, 2009.

Mazumder, R., Friedman, J. H., and Hastie, T. Sparsenet: coordinate descent with non-convex penalties. *Journal of the American Statistical Association*, 106:1125–1138, 2011.

Ogawa, K., Imamura, M., Takeuchi, I., and Sugiyama, M. Infinitesimal annealing for training semi-supervised support vector machines. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

Ritter, K. On parametric linear and quadratic programming problems. *mathematical Programming: Proceedings of the International Congress on Mathematical Programming*, pp. 307–335, 1984.

Shen, X., Tseng, G., Zhang, X., and Wong, W. H. On $\psi$-learning. *Journal of the American Statistical Association*, 98(463):724–734, 2003.

Vapnik, V. N. *The Nature of Statistical Learning Theory*. Springer, 1996.

Wu, Y. and Liu, Y. Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102:974–983, 2007.

Xu, L., Crammer, K., and Schuurmans, D. Robust support vector machine traiing via convex outlier ablation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2006.

Yu, Y., Yang, M., Xu, L., White, M., and Schuurmans, D. Relaxed clipping: a global training method for robust regression and classification. In *Advances in Neural Information Processing Systems*, volume 23, 2010.

Yuille, A. L. and Rangarajan, A. The concave-convex procedure (cccp). In *Advances in Neural Information Processing Systems*, volume 14, 2002.

Zhang, C. H. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38:894–942, 2010.

Zhou, H., Armagan, A., and Dunson, D. B. Path following and empirical Bayes model selection for sparse regression. *arXiv:1201.3528*, 2012.