Self-Training with Adaptive Regularization for S3VM

Edward Cheung
Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, N2L 3G1
Email: eycheung@uwaterloo.ca

Yuying Li
Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, N2L 3G1
Email: yuying@uwaterloo.ca

***Abstract*—The Semi-Supervised Support Vector Machine (S3VM) solves a non-convex, Mixed-Integer Program (MIP). Due to difficulty in solving the problem, convex approximations have typically been used. However, existing approaches suffer from poor scalability and struggle on certain datasets, compared to graph based counterparts. The poor predictive performance suggests that for some datasets, convex approximations may not be a sufficiently accurate approximation to the problem. We present a self-training approach with self-adapting regularization parameters for S3VM formulations. At each iteration, the regularization parameters are adapted to better reflect label confidence, class proportion, and to gradually include more unlabeled points. We show that updating the S3VM framework iteratively in this fashion, the sequence of SVM subproblems can be solved very efficiently and the solution generated by this sequence yields superior performance compared to leading SSL methods.**

I. Introduction

In many machine learning applications, unlabeled data is typically abundant. However, it is often very difficult to obtain labels for data. Usually, labels must be assigned manually, and this task is largely infeasible for most modern datasets. Semi-supervised learning (SSL) is a paradigm that incorporates both unlabeled and labeled data into the training process, allowing algorithms to leverage the abundant unlabeled data. An intuitive idea to extend the SVM framework to the SSL framework is to treat the unknown labels as optimization variables. By solving the SVM problem with the unlabeled data, a labeling can be found that encourages the decision boundary through low density regions. Approaches that follow this idea fall under the category of *Semi-Supervised SVM* (S3VM) [7].

Although S3VM is a natural mathematical formulation to incorporate unlabeled information, the optimization problem becomes a difficult Mixed-Integer Problem (MIP). Thus S3VM loses the desirable convex and continuous properties from the original SVM problem, and efficient optimization becomes difficult. A variety of approaches have been attempted, including local combinatorial searches [15], branch and bound techniques [6], semidefinite programming (SDP) [9], concave-convex procedures [8], and convex relaxations [17]. A full survey of the techniques can be found in [7].

The branch-and-bound (BB) approach proposed in [6] found that the global optimum found by the BB approach at times showed strong generalization performance, even when the traditional S3VM based methods typically struggled. This suggests that existing relaxation methods may not be a sufficiently accurate approximation to the original problem, and better approximations may lead to better generalization performance.

In this paper, we propose a **S**elf-**T**raining with **A**daptive **R**egularization SVM framework (STAR-SVM), which uses self-training to gradually incorporate unlabeled information. We view this as gradually approximating the original non-convex optimization problem, by a sequence of convex SVM subproblems, which can be readily solved. Additionally, we will show that the labeled set is grown in a way such that each subproblem should be quickly solved from the warm-start solution provided by the previous subproblem. This allows STAR-SVM to find an approximate solution to the non-convex optimization problem very quickly.

Since self-training can be misled by training errors and noisy datasets, we introduce individual regularization parameter C_i for the loss of each data point which automatically adapts to the degradation in confidence at each iteration. A confidence-weight parameter γ is utilized to control the rate at which the confidence declines.

II. Background

A. S3VM

The S3VM problem formulates as a binary classification problem where the training set is only partially labeled. Explicitly, we are given l labeled points $\{x_i, y_i\}_{i=1}^{l}, y_i = \pm 1$, and u unlabeled points, $\{x_i\}_{i=l+1}^{n}$, with $n = l + u$. We will use the sets $\mathcal{L}$ and $\mathcal{U}$ to denote the indices for labeled and unlabeled sets respectively. The following optimization problem is then solved to obtain the optimal hyperplane parameters (w, b), as well as the binary labels for the unlabeled set, $y_{\mathcal{U}} = [y_{l+1}, ..., y_n]^T$,

$$\begin{aligned} \min_{y_{\mathcal{U}}} \min_{w,b} \; & P(w, b, y_{\mathcal{U}}) = \frac{1}{2} w^T w + C \sum_{i \in \mathcal{L}} \xi_i^p + C^* \sum_{i \in \mathcal{U}} \xi_i^p \\ & s.t. \; y_i(w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \; i = 1, ..., n \\ & \xi_i \geq 0 \end{aligned} \tag{1}$$

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

where ϕ is a suitable feature map. For the remainder of this paper, we will use $p = 1$. For many approaches, the inner optimization in (1) is replaced with the dual as follows,

$$\begin{aligned} \min_{y_{\mathcal{U}}} \max_{\alpha} \; & D(\alpha, y_{\mathcal{U}}) = e^T\alpha - \frac{1}{2}\alpha^T Y K Y \alpha \\ & s.t.\; y^T\alpha = 0 \\ & \quad 0 \le \alpha_i \le C,\; i \in \mathcal{L} \\ & \quad 0 \le \alpha_j \le C^*,\; j \in \mathcal{U} \end{aligned} \tag{2}$$

where $Y = diag(y)$ and e is the all-ones vector. The resulting problem can be interpreted as finding the local maximizer with the smallest objective value amongst all local maxima. Since each possible labeling of $y_{\mathcal{U}}$ yields a different local maximum, there are an exponential number of maxima to search through. The combinatorial nature of $y_{\mathcal{U}}$ assignment makes solving either optimization problem extremely challenging.

B. Self-Training

Self-training is one of the earliest SSL techniques which uses a supervised algorithm to gradually expand the labeled set. Self-training starts by training solely on labeled data. At each iteration, an assessment is made on the predicted labels, using the current decision function for the unlabeled set. Points that are labeled confidently will be added to the labeled set and the supervised method is retrained on the enlarged labeled set, see e.g. [5]. An issue with self-training is that the intermediate supervised learning steps do not incorporate unlabeled information. Additionally, errors tend to propagate since each supervised learner assumes the labeled set is correct.

III. STAR-SVM

In this section, we will motivate the proposed STAR-SVM algorithm, demonstrating that by incrementally adjusting the regularization parameters, we can incorporate unlabeled information in a structured manner. Rather than iteratively increasing the impact of the entire unlabeled set as is done in annealing, we will gradually incorporate unlabeled examples to iteratively grow the labeled set. We will also give an overview of the algorithm and provide techniques to further speed up the method.

A. Motivation

The proposed method is motivated by the question of whether the struggle of S3VM is due to the formulation, or if instead it is from the convex relaxation. As noted in [6], the exact solution from the non-convex optimization problem can lead to significant improvements over existing convex relaxations.

We attempt to solve the combinatorial S3VM optimization problem by successively approximating the problem with convex SVM subproblems using only partial labels. The idea is to gradually grow the labeled set with self-training. Self-training for S3VM problems has been proposed before in [11, 12, 18]. However, none of these methods address a natural shortcoming of self-training, which is that errors tend to propagate since each iteration assumes the labeled set is correct. To address these issues, we introduce the idea of *adaptive regularization* into the self-training framework.

For most S3VM frameworks, there are separate regularization parameters, C and C^*, that bound the optimization variables for the labeled and unlabeled examples respectively. It is suggested to choose C^* to be smaller than C (usually $C^* = 0.1C$) to reflect that we are *less* confident in the unlabeled examples, [7]. Choosing a smaller constant penalizes the errors for the unlabeled set less and allow for less contribution to the decision hyperplane. Ideally, if we knew beforehand the confidence in each label, we could assign an individual C_i to each example to best reflect the label confidence and better reflect the problem. Utilizing this idea for an S3VM problem is difficult because the labels are unknown, and thus, confidence measures are also unknown. However, when self-training is used, the confidence in the labeling can be approximated by the iteration when the example is labeled. The idea is that earlier labels are more reliable than later labels due to error propagation. To reflect the confidence degrading at each iteration, we introduce a confidence weighting parameter, $\gamma \in [0, 1]$. Thus, when example x_i is labeled at iteration k, we adjust the regularization parameter C_i as follows,

$$C_i = \gamma^k C. \tag{3}$$

By limiting the contributions of examples we are less confident in, the effect of error propagation will be lessened.

1) Ignoring the Offset: The work in [19] has shown that when using kernels with a large feature space, such as the Gaussian RBF kernel, that utilizing the offset term, b, does not improve generalization performance for classification problems. Moreover, the effect of removing the offset term allows us to consider the optimization problem (2) without the equality constraint, allowing for simpler solvers such as the one in [20]. For the remainder of the paper, we will refer to this particular SVM solver as the *Training Without Offset SVM* (TWO-SVM), where Algorithm 1 (the 1D-SVM solver) is used. In this section, we will further illustrate that utilizing the offset term is not suitable for iterative semi-supervised learning procedures.

We train the SVM for a two moons problem with and without offset using default LibSVM parameters $C = \sigma = 1$ [4]. To demonstrate, in Figure 1 we see that the decision boundary obtained using offset conforms very tightly to the given labels. For some datasets, a reasonable decision boundary can be obtained through parameter tuning, but in many SSL problems, the labeled set is not large enough to allow for reasonable parameter tuning. The issue stems from the fact that the offset provides a global bias on the decision function, however, we generally have very limited information on the entire space. When using the offset term, we can see from Figure 1 that iteratively assigning new labels based on the decision function with offset can lead to very poor generalization performance. Thus, it seems more fitting for an iterative procedure such as the proposed method to use the offset-free version of the SVM problem.

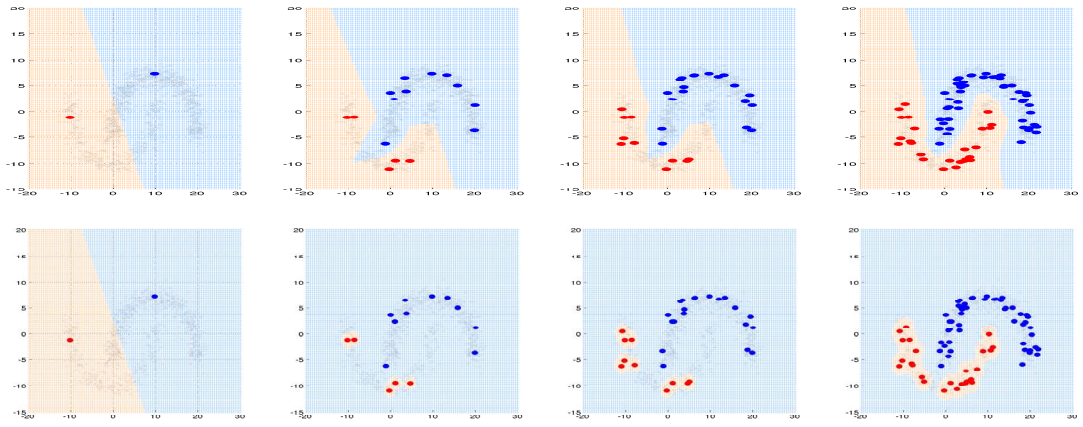

Fig. 1. We compare the decision boundaries returned from an SVM on several inputs with and without offset. The top row is trained without offset and the bottom row is trained with offset with $\sigma = 1$ and $C = 1$ (the default parameters for LibSVM). We see that when the labeled sets are small, the offset term b biases the decision boundary greatly and overfits the solution.

Consequently we will use the offset-free version of the S3VM problem in the proposed method.

2) Optimization Formulation: The proposed STAR-SVM framework solves a sequence of optimization problems with C_i^* values sequentially determined by the process. For notational convenience, rather than keeping track of indices in $\mathcal{U}$ which are newly labeled, the index sets will be updated at each iteration. That is, if example x_i is assigned a label at iteration k, then we update the index sets as follows,

$$\begin{aligned}\mathcal{L}_{k+1} &= \mathcal{L}_k \cup \{i\} \\ \mathcal{U}_{k+1} &= \mathcal{U}_k - \{i\}.\end{aligned}$$

We will use the convention that $\mathcal{L}_0$ and $\mathcal{U}_0$ will represent the initial labeled and unlabeled sets respectively.

The primal optimization problem for each STAR-SVM at iteration k is written as,

$$\begin{aligned}\min_{y_\mathcal{U}} \min_{w} \; & P_k(w, y_\mathcal{U}) = \frac{1}{2} w^T w + \sum_{i \in \mathcal{L}_k} C_i \xi_i + \sum_{i \in \mathcal{U}_k} C_i^* \xi_i \\ & s.t.\; y_i (w \cdot \phi(x_i)) \geq 1 - \xi_i,\; i = 1, ..., n \\ & \quad \xi_i \geq 0,\; i = 1, ..., n.\end{aligned} \tag{4}$$

For notational simplicity, we have dropped the dependencies of the regularization parameters C_i and C_i^* on k. The associated dual is written as,

$$\begin{aligned}\min_{y_\mathcal{U}} \max_{\alpha} \; & D_k(\alpha, y_\mathcal{U}) = e^T \alpha - \frac{1}{2} \alpha^T Y K Y \alpha \\ & s.t.\; 0 \leq \alpha_i \leq C_i,\; i \in \mathcal{L}_k \\ & \quad 0 \leq \alpha_j \leq C_j^*,\; j \in \mathcal{U}_k,\end{aligned} \tag{5}$$

where Y is the diagonal matrix of labels, and K is the kernel matrix.

Since no confidence or label information is assumed a priori for points in $\mathcal{U}_0$, we initialize $C_i^* = 0, \forall i \in \mathcal{U}_0$. Thus each subproblem involves training an SVM on the labeled set $\mathcal{L}_k$. In addition, the optimizations problem solved at each iteration are standard SVM problems rather than combinatorial optimization problems, which are intractable.

Note that by incorporating the class proportion information into the regularization parameters C_i and using the offset-free SVM formulation, the optimization problem does not have any equality constraint. This is an important point since it makes solving the SVM problem from a warm-start easier. Note that for other formulations, solving the SVM requires working sets with a minimum of two elements to preserve the equality constraints. The SVM subproblems for STAR-SVM can be solved using TWO-SVM, which exploits updating warm-start solutions.

3) Solving the SVM Subproblems: We give a brief overview of the TWO-SVM method proposed by [20]. As mentioned, without offset, the SVM problem can be solved using working sets of size one. This simplifies the training process allowing for a greedy component-wise gradient descent. At each iteration, an index i^* is identified which achieves the greatest improvement in the dual objective value. Let $\ell(x_i, y_i, \alpha)$ be the hinge-loss function

$$\ell(x_i, y_i, \alpha) := \max \left\{ 0, 1 - y_i \sum_{j \in \mathcal{L}_k} K(x_i, x_j) y_j \alpha_j \right\}. \tag{6}$$

The procedure is continued until the duality gap,

$$\begin{aligned}\text{gap}(\alpha) = \alpha^T Y K Y \alpha - e^T \alpha + \sum_{i \in \mathcal{L}_k} C_i \ell(x_i, y_i, \alpha) \\ + \sum_{i \in \mathcal{U}_k} C_i^* \ell(x_i, y_i, \alpha)\end{aligned} \tag{7}$$

becomes sufficiently small.

B. Confidence Score

For self-training to be successful, a reasonable confidence score must be chosen. The confidence score suggested for branching in [6] is the increase in the lower bound objective value if the label were swapped. However, this would involve

$2|\mathcal{U}_k|$ SVM solves at each iteration, which is not feasible for large datasets. In the following theorem, we will show that using the magnitude of the functional margin is a reasonable choice to assign confidence.

Without loss of generality, we will partition our variables into labeled and unlabeled set as follows,

$$K = \begin{pmatrix} K_{\mathcal{LL}} & K_{\mathcal{LU}} \\ K_{\mathcal{UL}} & K_{\mathcal{UU}} \end{pmatrix},\ y = \begin{pmatrix} y_{\mathcal{L}} \\ y_{\mathcal{U}} \end{pmatrix},\ \alpha = \begin{pmatrix} \alpha_{\mathcal{L}} \\ \alpha_{\mathcal{U}} \end{pmatrix}, \tag{8}$$

where $\mathcal{L}$ and $\mathcal{U}$ corresponds to the rows or columns indexed by $\mathcal{L}_k$ and $\mathcal{U}_k$, respectively.

Define the function,

$$\Delta D_k(\alpha, y|I) := D_k(\alpha, y) - D_k(\alpha, \tilde{y}) \tag{9}$$

where $\tilde{y}_j = y_j$ when $j \notin I$ and $\tilde{y}_j = -y_j$ when $j \in I$. $\Delta D(\alpha, y|I)$ is the difference between the objective values in (5) if the indices in I have their labels swapped.

Theorem 1. *Let $\alpha_{\mathcal{L}}$ be a feasible solution to the offset-free dual SVM problem* (5)*, trained on the labeled set $\mathcal{L}_k$ with corresponding labels $y_{\mathcal{L}}$. Let $\mathcal{L}_{k+1} = \mathcal{L}_k \cup \{i\}$, for any $i \in \mathcal{U}_k$. Then for any $\alpha_i \geq 0$, $\Delta D_{k+1}([\alpha_{\mathcal{L}}, \alpha_i], [y_{\mathcal{L}}, y_i]|\{i\}) \leq 0$ when $y_i = sign(\sum_{j \in \mathcal{L}_k} K_{i,j} y_j \alpha_j)$. Moreover, $\Delta D_k(\alpha, y|\{i\}) \propto |K_{i,\mathcal{L}} Y_{\mathcal{L}} \alpha_{\mathcal{L}}|$.*

Proof. Let $Y_{\mathcal{L}} = diag(y_{\mathcal{L}})$. Then we can partition D_{k+1} as follows.

$$\begin{aligned} D_{k+1}([\alpha_{\mathcal{L}}, \alpha_i], y_i) = -\frac{1}{2}\alpha_{\mathcal{L}}^T Y_{\mathcal{L}} K_{\mathcal{LL}} Y_{\mathcal{L}} \alpha_{\mathcal{L}} + e^T \alpha_{\mathcal{L}} - \\ \alpha_i y_i K_{i,\mathcal{L}} Y_{\mathcal{L}} \alpha_{\mathcal{L}} - \frac{1}{2}\alpha_i^2 K_{ii} + \alpha_i \end{aligned} \tag{10}$$

Assigning $y_i = sign(\sum_{j \in \mathcal{L}_k} K_{i,j} y_j \alpha_j)$, we have

$$\begin{aligned} D_{k+1}([\alpha_{\mathcal{L}}, \alpha_i], [y_{\mathcal{L}}, y_i]) - D_{k+1}([\alpha_{\mathcal{L}}, \alpha_i], [y_{\mathcal{L}}, -y_i]) = \\ -2\alpha_i y_i K_{i,\mathcal{L}} Y_{\mathcal{L}} \alpha_{\mathcal{L}} \end{aligned} \tag{11}$$

Since $sign(y_i) = sign(K_{i,\mathcal{L}} Y_{\mathcal{L}} \alpha_{\mathcal{L}})$ and $\alpha_i \geq 0$, the quantity $-2\alpha_i y_i K_{i,\mathcal{L}} Y_{\mathcal{L}} \alpha_{\mathcal{L}} \leq 0$. Thus,

$$\Delta D_{k+1} \leq 0 \tag{12}$$

Also from (11), we see that

$$\Delta D_k(\alpha, y|\{i\}) \propto |K_{i,\mathcal{L}} Y_{\mathcal{L}} \alpha_{\mathcal{L}}|. \tag{13}$$

□

Recall that S3VM finds the local maximizer with the smallest objective value. Theorem 1 states we can decrease the objective value by simply swapping labels for any points that are inconsistent with the decision hyperplane. Also, we have shown that the magnitude of the functional margin is proportional to the change from swapping labels. Thus, choosing examples that have the largest functional margin corresponds to finding the coordinate which yields the largest rate of change in ΔD_k.

This relationship only considers the solutions after updating α_i and keeping $\alpha_{\mathcal{L}}$ constant. If this solution is not optimal, the variables in $\alpha_{\mathcal{L}}$ will also need to be updated. However, using a warm-start solution with TWO-SVM, the solution after the 1D update to α_i can be used as an approximate solution to (5). We quantify the quality of this approximation using the duality gap and Theorem 2 stated below.

Theorem 2. *Let $\alpha^*_{\mathcal{L}}$ be the optimal solution the SVM dual problem* (5) *trained on $\mathcal{L}_k$ with corresponding labels $y^{(k)}$. Let α^* be the optimal solution for problem* (5) *trained on $\mathcal{L}_{k+1} = \mathcal{L}_k \cup \{i\}$ with labels $y^{(k+1)}$. Define the dual suboptimality measure as $\nu(\alpha) = D(\alpha^*, y) - D(\alpha, y)$, where α is any feasible solution. If α_i is the optimal 1D update from a warm-start solution $\alpha = [\alpha^*_{\mathcal{L}}, 0]$, given by an iteration of TWO-SVM, then $\nu(\alpha) \leq C_i^* \ell(x_i, y_i, \alpha)$, where ℓ is the hinge-loss function defined in* (6)*.*

Proof. As noted in [20], the duality gap for a feasible solution α is given by,

$$\begin{aligned} \text{gap}(\alpha) = \alpha^T Y K Y \alpha - e^T \alpha + \sum_{i \in \mathcal{L}_k} C_i \ell(x_i, y_i, f) \\ + \sum_{i \in \mathcal{U}_k} C_i^* \ell(x_i, y_i, f) \end{aligned} \tag{14}$$

By strong duality, since $\alpha^*_{\mathcal{L}}$ is the optimal solution to the SVM problem trained on $\mathcal{L}_k$, $\text{gap}(\alpha^*_{\mathcal{L}}) = 0$. Thus, using the warm start solution for $\mathcal{L}_{k+1}$, we have that,

$$\text{gap}([\alpha^*_{\mathcal{L}}, 0]) = C_i \ell(x_i, y_i, f) \tag{15}$$

Consider,

$$\nu([\alpha^*_{\mathcal{L}}, \alpha_i]) = D_{k+1}(\alpha^*, y^{(k+1)}) - D_{k+1}([\alpha^*_{\mathcal{L}}, \alpha_i], y^{(k+1)}) \tag{16}$$

where $\alpha^*_{\mathcal{L}}$ and α^* are the optimal solutions trained on $(\mathcal{L}_k, y^{(k)})$ and $(\mathcal{L}_{k+1}, y^{(k+1)})$ respectively. Let $P_{k+1}(w^*, y^{(k+1)})$ be the optimal value to the primal problem for the training set $(\mathcal{L}_{k+1}, y^{(k+1)})$. Let $(w', y^{(k+1)})$ be the primal solution associated with the dual solution $([\alpha^*_{\mathcal{L}}, 0], y^{(k+1)})$, where $\alpha^*_{\mathcal{L}}$ is the optimal solution given by TWO-SVM for the training set $(\mathcal{L}_k, y^{(k)})$, with $y_i^{(k)} = y_i^{(k+1)}$ for all $i \in \mathcal{L}_k$. Then,

$$\begin{aligned} & P_{k+1}(w^*, y^{(k+1)}) \\ & \leq P_{k+1}(w', y^{(k+1)}) \\ & = D_{k+1}([\alpha^*_{\mathcal{L}}, 0], y^{(k+1)}) + \text{gap}([\alpha^*_{\mathcal{L}}, 0]). \end{aligned} \tag{17}$$

Again from strong duality, we have that $P_{k+1}(w^*, y^{(k+1)}) = D_{k+1}(\alpha^*, y^{(k+1)})$. Substituting (17) into (16), we get,

$$\begin{aligned} & \nu([\alpha^*_{\mathcal{L}}, \alpha_i]) \\ & \leq D_{k+1}([\alpha^*_{\mathcal{L}}, 0], y^{(k+1)}) + \text{gap}([\alpha^*, 0]) - D_{k+1}([\alpha^*_{\mathcal{L}}, \alpha_i], y^{(k+1)}) \\ & \leq \text{gap}([\alpha^*, 0]) \end{aligned} \tag{18}$$

where the last inequality comes from the fact that TWO-SVM updates α_i to increase the objective function, so $D_{k+1}([\alpha^*_{\mathcal{L}}, \alpha_i], y^{(k+1)}) \geq D_{k+1}([\alpha^*_{\mathcal{L}}, 0], y^{(k+1)})$. This completes the proof. □

This indicates that after one step of the TWO-SVM algorithm, the dual suboptimality measure will be bounded above

by a small number. This implies that the warm-start solution is very close to the optimal solution, and the TWO-SVM algorithm will converge very quickly. This is useful since this will practically reduce the runtime, but it also indicates the new label is consistent with the existing classifier, since it does not lead to a large perturbation in the solution, which is a desirable property in the context to self-training.

Moreover, Lemma 3 from [20] states a relation between the duality gap and the step sizes of the ascent directions.

Lemma 3. *(Steinwart et al. [20]) Let* $\alpha \in [0, C] = [0, C_1] \times [0, C_2] \times ... \times [0, C_n]$ *and define the function,* $\sigma : (\alpha, I \subseteq \{1, ..., n\}) \to \mathbb{R}$ *as,*

$$\sigma(\alpha|I) := \sup_{\substack{\tilde{\alpha}\in[0,C] \\ \tilde{\alpha}_i=\alpha_i, \forall i \notin I}} \langle \nabla D(\alpha), \tilde{\alpha} - \alpha \rangle \tag{19}$$

where $\nabla D(\alpha)$ *is the gradient of the SVM dual objective function in* (2) *evaluated at a feasible point* α*. Then,*

$$\sum_{i=1}^{n} \sigma(\alpha|\{i\}) = gap(\alpha). \tag{20}$$

From this lemma we see that by choosing examples that yield small duality gaps, we limit the sum of the ascent direction step sizes. This means that we can find solutions that are closer to the warm-start, allowing for a smooth labeling.

C. Computational Efficiency

In practice, it is inefficient to only add one label at each iteration, since there are likely several examples which are quite similar and can be added all at once without compromising the performance of the algorithm. In most self-training procedures, a confidence threshold is chosen to accept all labelings that are above this threshold. To avoid introducing new hyperparameters, we propose to use an adaptive thresholding scheme in Algorithm 1. The idea is to threshold the confidence scores, where we gradually decrease this threshold as more points become labeled.

Algorithm 1 Adaptive Thresholding

Input: confidence scores g, labels assigned so far by STAR-SVM l, original number of unlabeled points u_0
Output: indices of confident labels I
$\tau \leftarrow \frac{l}{u_0}$
$I^+ \leftarrow \{i : g_i \geq (1-\tau)\max(g)\}$
$I^- \leftarrow \{i : -g_i \geq (1-\tau)\max(-g)\}$
$I \leftarrow I^+ \cup I^-$

1) Incorporating Class Proportion Knowledge: For imbalanced datasets, penalizing the regularization parameters, C, for the positive and negative classes separately can improve performance, see, e.g. [14]. When data is highly imbalanced, the positive (minority class) labels may be ignored by the classifier since the loss associated will be small due to the small size of the minority class. Thus, by adjusting the regularization parameters to reflect the class ratio, we also penalize the loss accordingly. For our iterative approach, we can adjust the regularization parameters accordingly to reflect the prior knowledge of the class ratios when the labels are assigned. Let r be the proportion of positive examples (assuming the positive class is the minority class), either provided as a user parameter or estimated from the labeled data, $r = \frac{\sum_{i=1}^{l} max(0, y_i)}{l}$, where $l = |\mathcal{L}_0|$. Then the associated weights will be defined as,

$$\begin{aligned} C^+ &= rC \\ C^- &= (1-r)C. \end{aligned} \tag{21}$$

Combining this with (3), the update for C_i at iteration k becomes,

$$C_i = \begin{cases} \gamma^k C^+, & \text{if } y_i = 1 \\ \gamma^k C^-, & \text{if } y_i = -1. \end{cases} \tag{22}$$

D. Algorithm Overview

We now briefly summarize the proposed algorithm, STAR-SVM, in Algorithm 2.

Algorithm 2 STAR-SVM

Input: data $\mathcal{X}$, labeled indices $\mathcal{L}_0$, unlabeled indices $\mathcal{U}_0$, labels y, kernel matrix K, regularization parameter C, confidence weight γ, class proportion r
Output: $f^*(x) = \sum_{i=1}^{n} K(x, x_i) y_i \alpha_i$
$k \leftarrow 1$
repeat
  $\alpha \leftarrow \texttt{TWO-SVM}(\mathcal{X}_\mathcal{L}, y_\mathcal{L}, C_\mathcal{L})$
  $g \leftarrow K_{\mathcal{L}\mathcal{L}} diag(y_\mathcal{L}) \alpha$
  $T \leftarrow \texttt{threshold}(g, |\mathcal{L}_k| - |\mathcal{L}_0|, |\mathcal{U}_0|)$
  for $j \in T$ **do**
    if $g_j > 0$ **then**
      $C_j \leftarrow \gamma^k C^+$
    else
      $C_j \leftarrow \gamma^k C^-$
    end if
  end for
  $\mathcal{U}_{k+1} \leftarrow \mathcal{U}_k - T$
  $y_T \leftarrow sign(g_T)$
  $\mathcal{L}_{k+1} \leftarrow \mathcal{L}_k \cup T$
  $k \leftarrow k + 1$
until $\mathcal{U}_k = \emptyset$

Although STAR-SVM in Algorithm 2 is motivated in the binary class setting, extending this algorithm to a multi-class algorithm can be accomplished using a standard one-vs-rest approach. If there are c classes, we train the STAR-SVM algorithm c times where each trial corresponds to a different class being the positive class. An unlabeled example x_i, is given the label j, if x_i was labeled positive in the earliest iteration for the trial where j the positive class (or the latest trial j if x_i is labeled negatively for all trials). This labeling agrees with the assumption that earlier labellings correspond to higher confidence. Although other approaches can be considered, this approach will be used for the subsequent experimental results.

TABLE I
IMBALANCED DATSET DESCRIPTION

Dataset Name	Source	Dimension	Points	Num. Positive
Ecoli	UCI	7	336	35
Crime	UCI	100	1994	100
Libras	UCI	7	360	90
Oil	[16]	49	937	41
Optical Digits	UCI	64	5620	554
Wine	UCI	11	4898	183
Satellite Image	UCI	36	6435	626
Vowel	Keel	10	989	90
Shuttle0vs4	Keel	9	1829	123
Page Block	Keel	10	5472	559
Glass4	Keel	9	214	13
Clevelan0vs4	Keel	13	173	13
Contraceptive	Keel	9	1473	333
Euthyroid	UCI	42	3163	293
Spectrometer	UCI	93	531	45

IV. EXPERIMENTAL RESULTS

In this section, we compare STAR-SVM against leading SSL methods on a variety of imbalanced datasets, both graph and S3VM based methods are considered. The benchmark datasets are provided by UCI [2] and Keel [1]. A wide variety of datasets are chosen to best reflect varying structural properties of the data as well as varying class proportion ratios. This experiment will examine the performance of the classifiers in a setting where little is known about the true distributions of the data. We emphasize imbalanced datasets since this is an area where SSL methods typically struggle the most [21, 22] and to emphasize the importance of adapting regularization parameters to incorporate class imbalances. Additionally, since S3VM methods are inherently binary classifiers, any multi-class setting will typically require solving several imbalanced S3VM problems.

A. Imbalanced Datasets

We demonstrate STAR-SVM's performance across a variety of imbalanced realworld datasets from UCI [2] and Keel [1]. The multi-class data has been transformed into imbalanced binary datasets using the classes suggested in [10]. We summarize the descriptions in Table I.

The methods we compare against are,

1) Gaussian Fields and Harmonic Functions (GFHF) [24], using the harmonic solution to solve the graph diffusion.
2) Local and Global Consistency (LGC) [23], using normalized graph Laplacian and soft constraints for given labels.
3) Greedy Gradient Maximum Cut (GGMC) [21, 22], alternating minimization over both discrete labels and continuous decision function.
4) Laplacian SVM (LapSVM) [3], adding graph regularization term to SVM framework.
5) WellSVM [17], convex relaxtion to S3VM.

The code for each method is provided by the authors.

For most SSL problems, the labeled set is insufficient to perform reliable parameter tuning. Moreover, as noted in [7], some methods benefit more from parameter tuning than others. Thus, to ensure fairness and to simulate a realistic scenario, parameter tuning is not applied to the datasets in Table I. Instead, all methods used the same set of parameters summarized in the Table II.

TABLE II
PARAMETERS

Parameter	Value	Description
k	6	k used for k-NN graph construction
kernel type	RBF	kernel function used
σ	1	RBF kernel width
γ_A	1	kernel regularization weight for LapSVM
γ_I	0.1	graph regularization weight for LapSVM
γ	0.7	confidence weight for STAR-SVM
C	1	SVM regularization parameter

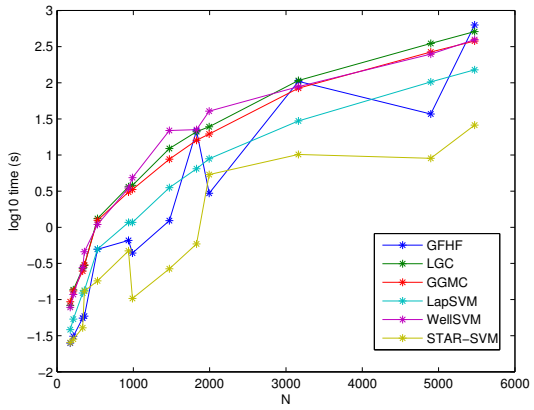

Fig. 2. CPU times for imbalanced datasets.

For LapSVM, the results reported here ignore the offset term as well. Despite the fact that the original LapSVM solver includes a bias term, we have found that the results for LapSVM have improved dramatically when the offset is ignored for the classification.

For each dataset, $n = 20$ labeled samples are randomly chosen, with at least one labeled example from each class. For each dataset, the experiments are repeated using thirty different random initial labeled sets. The imbalance ratio r is estimated from the given labels. The average performances are reported.

For imbalanced datasets, we will report the F_1-score, Geometric Mean, and AUC which are more suitable for imbalanced classification [13].

B. Time Comparisons

For the experiments run on the imbalanced datasets, we report the average CPU time required to run each algorithm. All algorithms are implemented in MATLAB, however, LapSVM and WellSVM utilize subroutines written in C++.

In Figure 2, we see that STAR-SVM scales much better as the datasets increase in size. Experimentally, after the first few iterations, each TWO-SVM solve terminates quickly.

TABLE III
F_1 SCORE WITH 20 LABELED EXAMPLES. THE TOP PERFORMERS ARE BOLDED IN EACH ROW USING A t-TEST WITH A 95% SIGNIFICANCE LEVEL.

Dataset Name	GFHF	LGC	GGMC	LapSVM	WellSVM	STAR-SVM
Ecoli	$\mathbf{0.54 \pm 0.18}$	0.05 ± 0.17	$\mathbf{0.48 \pm 0.05}$	$\mathbf{0.50 \pm 0.15}$	$\mathbf{0.48 \pm 0.19}$	$\mathbf{0.53 \pm 0.05}$
Crime	0.12 ± 0.06	0.15 ± 0.10	0.19 ± 0.07	0.21 ± 0.08	$\mathbf{0.35 \pm 0.14}$	$\mathbf{0.29 \pm 0.10}$
Libras	$\mathbf{0.40 \pm 0.18}$	$\mathbf{0.49 \pm 0.17}$	0.26 ± 0.10	$\mathbf{0.47 \pm 0.19}$	0.31 ± 0.23	$\mathbf{0.49 \pm 0.16}$
Oil	0.08 ± 0.01	$\mathbf{0.17 \pm 0.13}$	0.12 ± 0.09	$\mathbf{0.16 \pm 0.13}$	0.08 ± 0.05	$\mathbf{0.22 \pm 0.14}$
Optical Digits	0.01 ± 0.01	$\mathbf{0.72 \pm 0.21}$	0.39 ± 0.12	$\mathbf{0.64 \pm 0.17}$	0.23 ± 0.24	0.47 ± 0.08
Wine	0.01 ± 0.02	0.02 ± 0.02	0.07 ± 0.02	$\mathbf{0.10 \pm 0.06}$	$\mathbf{0.14 \pm 0.08}$	$\mathbf{0.10 \pm 0.05}$
Satellite Image	0.08 ± 0.12	0.28 ± 0.22	0.28 ± 0.07	0.36 ± 0.08	0.24 ± 0.23	$\mathbf{0.44 \pm 0.11}$
Vowel	$\mathbf{0.49 \pm 0.13}$	$\mathbf{0.48 \pm 0.16}$	0.41 ± 0.12	$\mathbf{0.55 \pm 0.09}$	$\mathbf{0.57 \pm 0.13}$	$\mathbf{0.51 \pm 0.09}$
Shuttle0vs4	0.87 ± 0.14	0.88 ± 0.16	0.94 ± 0.01	0.89 ± 0.09	0.76 ± 0.16	$\mathbf{1.00 \pm 0.00}$
Page Block	0.47 ± 0.20	0.50 ± 0.17	0.24 ± 0.07	$\mathbf{0.60 \pm 0.08}$	0.43 ± 0.22	$\mathbf{0.52 \pm 0.14}$
Glass4	$\mathbf{0.42 \pm 0.11}$	0.06 ± 0.10	0.40 ± 0.11	$\mathbf{0.45 \pm 0.11}$	0.35 ± 0.17	$\mathbf{0.49 \pm 0.11}$
Clevelan0vs4	$\mathbf{0.41 \pm 0.26}$	0.17 ± 0.20	$\mathbf{0.40 \pm 0.24}$	$\mathbf{0.44 \pm 0.22}$	$\mathbf{0.40 \pm 0.28}$	$\mathbf{0.50 \pm 0.21}$
Contraceptive	0.21 ± 0.08	0.18 ± 0.09	0.29 ± 0.05	0.32 ± 0.05	0.29 ± 0.08	$\mathbf{0.36 \pm 0.04}$
Euthyroid	0.13 ± 0.08	0.13 ± 0.09	0.17 ± 0.05	$\mathbf{0.22 \pm 0.07}$	0.07 ± 0.06	$\mathbf{0.23 \pm 0.06}$
Spectrometer	$\mathbf{0.55 \pm 0.19}$	0.37 ± 0.28	0.39 ± 0.19	0.50 ± 0.18	0.40 ± 0.15	$\mathbf{0.62 \pm 0.15}$
Average	0.319	0.310	0.335	0.428	0.339	**0.451**
Avg. Rank	4.267	4.267	4.400	2.400	4.067	**1.600**

TABLE IV
G-MEAN FOR 20 LABELED EXAMPLES

Dataset Name	GFHF	LGC	GGMC	LapSVM	WellSVM	STAR-SVM
Ecoli	0.72 ± 0.21	0.07 ± 0.21	0.82 ± 0.10	$\mathbf{0.76 \pm 0.26}$	0.67 ± 0.22	$\mathbf{0.87 \pm 0.02}$
Crime	0.26 ± 0.09	0.32 ± 0.15	$\mathbf{0.49 \pm 0.20}$	$\mathbf{0.47 \pm 0.19}$	$\mathbf{0.59 \pm 0.21}$	$\mathbf{0.53 \pm 0.13}$
Libras	$\mathbf{0.59 \pm 0.24}$	$\mathbf{0.67 \pm 0.22}$	$\mathbf{0.66 \pm 0.18}$	$\mathbf{0.70 \pm 0.24}$	0.49 ± 0.26	$\mathbf{0.68 \pm 0.19}$
Oil	0.03 ± 0.10	0.32 ± 0.17	$\mathbf{0.41 \pm 0.16}$	$\mathbf{0.39 \pm 0.19}$	0.28 ± 0.12	$\mathbf{0.47 \pm 0.13}$
Optical Digits	0.07 ± 0.02	$\mathbf{0.87 \pm 0.17}$	0.77 ± 0.11	$\mathbf{0.91 \pm 0.05}$	0.39 ± 0.29	0.78 ± 0.07
Wine	0.06 ± 0.07	0.07 ± 0.07	$\mathbf{0.49 \pm 0.07}$	0.39 ± 0.14	0.38 ± 0.14	$\mathbf{0.44 \pm 0.13}$
Satellite Image	0.16 ± 0.16	0.43 ± 0.27	$\mathbf{0.67 \pm 0.12}$	$\mathbf{0.75 \pm 0.13}$	0.42 ± 0.34	$\mathbf{0.76 \pm 0.14}$
Vowel	0.66 ± 0.15	0.64 ± 0.18	$\mathbf{0.78 \pm 0.11}$	$\mathbf{0.83 \pm 0.10}$	$\mathbf{0.75 \pm 0.14}$	0.76 ± 0.10
Shuttle0vs4	0.91 ± 0.13	$\mathbf{0.94 \pm 0.15}$	$\mathbf{1.00 \pm 0.00}$	0.90 ± 0.08	0.91 ± 0.07	$\mathbf{1.00 \pm 0.00}$
Page Block	0.58 ± 0.19	0.63 ± 0.18	0.60 ± 0.10	$\mathbf{0.81 \pm 0.08}$	0.62 ± 0.22	0.69 ± 0.14
Glass4	0.66 ± 0.14	0.10 ± 0.17	0.75 ± 0.09	$\mathbf{0.78 \pm 0.12}$	0.59 ± 0.22	$\mathbf{0.82 \pm 0.10}$
Clevelan0vs4	$\mathbf{0.56 \pm 0.32}$	0.24 ± 0.26	$\mathbf{0.63 \pm 0.34}$	$\mathbf{0.68 \pm 0.32}$	$\mathbf{0.53 \pm 0.33}$	$\mathbf{0.71 \pm 0.28}$
Contraceptive	0.38 ± 0.09	0.34 ± 0.11	0.48 ± 0.06	0.51 ± 0.06	0.47 ± 0.09	$\mathbf{0.55 \pm 0.04}$
Euthyroid	0.37 ± 0.15	0.29 ± 0.15	0.46 ± 0.12	$\mathbf{0.51 \pm 0.12}$	0.25 ± 0.12	$\mathbf{0.56 \pm 0.13}$
Spectrometer	$\mathbf{0.67 \pm 0.18}$	0.44 ± 0.29	$\mathbf{0.63 \pm 0.22}$	$\mathbf{0.70 \pm 0.18}$	0.62 ± 0.14	$\mathbf{0.75 \pm 0.14}$
Average	0.445	0.425	0.643	0.673	0.531	**0.691**
Avg. Rank	4.933	4.667	2.933	2.333	4.600	**1.533**

TABLE V
AUC FOR 20 LABELED EXAMPLES

Dataset Name	GFHF	LGC	GGMC	LapSVM	WellSVM	STAR-SVM
Ecoli	0.91 ± 0.02	0.67 ± 0.19	0.90 ± 0.01	0.92 ± 0.03	$\mathbf{0.93 \pm 0.04}$	$\mathbf{0.94 \pm 0.02}$
Crime	0.70 ± 0.10	0.73 ± 0.04	0.32 ± 0.04	0.69 ± 0.12	$\mathbf{0.85 \pm 0.08}$	0.76 ± 0.05
Libras	$\mathbf{0.85 \pm 0.18}$	$\mathbf{0.87 \pm 0.08}$	0.70 ± 0.12	$\mathbf{0.86 \pm 0.11}$	$\mathbf{0.81 \pm 0.10}$	$\mathbf{0.86 \pm 0.08}$
Oil	0.50 ± 0.02	$\mathbf{0.76 \pm 0.06}$	0.48 ± 0.05	$\mathbf{0.72 \pm 0.10}$	0.72 ± 0.05	$\mathbf{0.76 \pm 0.04}$
Optical Digits	0.69 ± 0.18	0.95 ± 0.06	0.95 ± 0.01	$\mathbf{0.98 \pm 0.02}$	0.72 ± 0.18	0.84 ± 0.07
Wine	0.56 ± 0.09	$\mathbf{0.67 \pm 0.05}$	0.42 ± 0.07	$\mathbf{0.65 \pm 0.08}$	$\mathbf{0.69 \pm 0.07}$	$\mathbf{0.66 \pm 0.07}$
Satellite Image	$\mathbf{0.88 \pm 0.05}$	0.64 ± 0.16	$\mathbf{0.88 \pm 0.04}$	$\mathbf{0.88 \pm 0.06}$	0.68 ± 0.21	0.79 ± 0.10
Vowel	$\mathbf{0.92 \pm 0.07}$	0.86 ± 0.09	0.88 ± 0.08	$\mathbf{0.92 \pm 0.05}$	$\mathbf{0.93 \pm 0.04}$	0.89 ± 0.05
Shuttle0vs4	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	0.98 ± 0.01	$\mathbf{1.00 \pm 0.00}$	0.99 ± 0.02	$\mathbf{1.00 \pm 0.00}$
Page Block	$\mathbf{0.90 \pm 0.05}$	0.87 ± 0.05	0.75 ± 0.11	$\mathbf{0.92 \pm 0.03}$	0.85 ± 0.07	$\mathbf{0.90 \pm 0.04}$
Glass4	0.88 ± 0.09	0.87 ± 0.03	0.66 ± 0.11	$\mathbf{0.95 \pm 0.05}$	0.93 ± 0.04	$\mathbf{0.96 \pm 0.02}$
Clevelan0vs4	0.92 ± 0.06	0.89 ± 0.02	0.56 ± 0.19	0.92 ± 0.06	$\mathbf{0.96 \pm 0.02}$	0.93 ± 0.04
Contraceptive	0.53 ± 0.05	0.53 ± 0.04	0.54 ± 0.04	0.53 ± 0.05	$\mathbf{0.59 \pm 0.07}$	$\mathbf{0.57 \pm 0.05}$
Euthyroid	$\mathbf{0.57 \pm 0.07}$	0.50 ± 0.06	$\mathbf{0.62 \pm 0.05}$	0.55 ± 0.07	0.57 ± 0.06	$\mathbf{0.57 \pm 0.09}$
Spectrometer	0.73 ± 0.13	0.84 ± 0.09	0.44 ± 0.06	$\mathbf{0.92 \pm 0.07}$	$\mathbf{0.88 \pm 0.06}$	$\mathbf{0.92 \pm 0.08}$
Average	0.769	0.776	0.672	**0.828**	0.808	0.823
Avg. Rank	3.933	4.133	4.733	2.867	2.933	**2.400**

C. Discussion

As observed in Tables III-V, STAR-SVM shows very strong performance across all metrics. We observe that for most datasets, one of the graph-based methods (GFHF, LGC, and GGMC) typically struggles. This highlights the sensitivity of the graph construction, which can be very challenging to tune correctly with very limited data.

V. Conclusions

In this paper, we propose a method STAR-SVM that is competitive against state-of-the-art SSL methods on various real-world datasets. We have shown that by gradually adjusting the regularization parameters, the optimization problems can adapt to incorporate structure from the unlabeled data even in the presence of noisy datasets. Also we have shown theoretically and experimentally that growing the labeled set in this fashion allows for smooth variations of the classifier, yielding very quick convergence in solving SVM subproblems using a warm-start. The strong performance of STAR-SVM, in a setting where little is known about the data, suggests that non-convex objective functions can be a promising direction for future S3VM algorithms.

References

[1] J Alcalá, A Fernández, J Luengo, J Derrac, S García, L Sánchez, and F Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(255-287):11, 2010.

[2] K. Bache and M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

[3] Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In *Learning theory*, pages 624–638. Springer, 2004.

[4] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/ cjlin/libsvm`.

[5] Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al. *Semi-supervised learning*, volume 2. MIT press Cambridge, 2006.

[6] Olivier Chapelle, Vikas Sindhwani, and S Sathiya Keerthi. Branch and bound for semi-supervised support vector machines. In *Advances in neural information processing systems*, pages 217–224, 2006.

[7] Olivier Chapelle, Vikas Sindhwani, and Sathiya S Keerthi. Optimization techniques for semi-supervised support vector machines. *The Journal of Machine Learning Research*, 9:203–233, 2008.

[8] Ronan Collobert, Fabian Sinz, Jason Weston, and Léon Bottou. Large scale transductive svms. *The Journal of Machine Learning Research*, 7:1687–1712, 2006.

[9] Tijl De Bie and Nello Cristianini. Semi-supervised learning using semi-definite programming. *Semi-supervised learning. MIT Press, Cambridge-Massachussets*, 32, 2006.

[10] Zejin Ding. Diversified ensemble classifiers for highly imbalanced data learning and their application in bioinformatics. 2011.

[11] Wael Emara, Mehmed Kantardzic Marcel Karnstedt, Kai-Uwe Sattler, Dirk Habich, and Wolfgang Lehner. An approach for incremental semi-supervised svm. In *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, pages 539–544. IEEE, 2007.

[12] Fei Gao, Jingyuan Mei, Jinping Sun, Jun Wang, Erfu Yang, and Amir Hussain. A novel classification algorithm based on incremental semi-supervised support vector machine. *PloS one*, 10(8):e0135709, 2015.

[13] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on*, 21(9):1263–1284, 2009.

[14] He He and Ali Ghodsi. Rare class classification by support vector machine. In *ICPR*, pages 548–551, 2010.

[15] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209, 1999.

[16] Miroslav Kubat, Robert C Holte, and Stan Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine learning*, 30(2-3):195–215, 1998.

[17] Yu-Feng Li, Ivor W Tsang, James T Kwok, and Zhi-Hua Zhou. Convex and scalable weakly labeled svms. *The Journal of Machine Learning Research*, 14(1):2151–2188, 2013.

[18] Yuanqing Li, Cuntai Guan, Huiqi Li, and Zhengyang Chin. A self-training semi-supervised svm algorithm and its application in an eeg-based brain computer interface speller system. *Pattern Recognition Letters*, 29(9):1285–1294, 2008.

[19] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[20] Ingo Steinwart, Don Hush, and Clint Scovel. Training svms without offset. *The Journal of Machine Learning Research*, 12:141–202, 2011.

[21] Jun Wang, Tony Jebara, and Shih-Fu Chang. Graph transduction via alternating minimization. In *Proceedings of the 25th international conference on Machine learning*, pages 1144–1151. ACM, 2008.

[22] Jun Wang, Tony Jebara, and Shih-Fu Chang. Semi-supervised learning using greedy max-cut. *The Journal of Machine Learning Research*, 14(1):771–800, 2013.

[23] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NIPS*, volume 16, pages 321–328, 2003.

[24] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003.