β-risk: a New Surrogate Risk for Learning from Weakly Labeled Data

Valentina Zantedeschi[*] **Rémi Emonet**

Marc Sebban

firstname.lastname@univ-st-etienne.fr
Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School,
Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France

Abstract

During the past few years, the machine learning community has paid attention to developing new methods for learning from weakly labeled data. This field covers different settings like semi-supervised learning, learning with label proportions, multi-instance learning, noise-tolerant learning, etc. This paper presents a generic framework to deal with these weakly labeled scenarios. We introduce the β-risk as a generalized formulation of the standard empirical risk based on surrogate margin-based loss functions. This risk allows us to express the reliability on the labels and to derive different kinds of learning algorithms. We specifically focus on SVMs and propose a soft margin β-SVM algorithm which behaves better that the state of the art.

1 Introduction

The growing amount of data available nowadays allowed us to increase the confidence in the models induced by machine learning methods. On the other hand, it also caused several issues, especially in supervised classification, regarding the availability of labels and their reliability. Because it may be expensive and tricky to assign a reliable and unique label to each training instance, the data at our disposal for the application at hand are often *weakly labeled*. Learning from weak supervision has received important attention over the past few years [14, 12]. This research field includes different settings: only a fraction of the labels are known (Semi-Supervised learning [22]); we can access only the proportions of the classes (Learning with Label Proportions [19] and Multi-Instance Learning [8]); the labels are uncertain or noisy (Noise-Tolerant Learning [1, 18, 16]); different discording labels are given to the same instance by different experts (Multi-Expert Learning [21]); labels are completely unknown (Unsupervised Learning [11]). As a consequence of this statement of fact, the data provided in all these situations cannot be fully exploited using supervised techniques, at the risk of drastically reducing the performance of the learned models. To address this issue, numerous machine learning methods have been developed to deal with each of the previous specific situations. However, all these weakly labeled learning tasks share common features mainly relying on the confidence in the labels, opening the door to the development of generic frameworks. Unfortunately, only a few attempts have tried to address several settings with the same approach. The most interesting one has been presented in [14] where the authors propose WELLSVM which is dedicated to deal with three different weakly labeled learning scenarios: semi-supervised learning, multi-instance learning and clustering. However, WELLSVM focuses specifically on Support Vector Machines and it requires to

[*] http://vzantedeschi.com/

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

derive a new optimization problem for each new task. Even though WELLSVM constitutes a step further towards general models, it stopped in midstream constraining the learner to use SVMs.

This paper aims to bridge this gap by presenting a generic framework for learning from weakly labeled data. Our approach is based on the derivation of the β-risk , a new surrogate empirical risk defined as a strict generalization of the standard empirical risk relying on surrogate margin-based loss functions. The main interesting property of the β-risk comes from its ability to exploit the information given by the weakly supervised setting and encoded as a β matrix reflecting the supervision on the labels. Moreover, the instance-specific weights β let one integrate in classical methods the side information provided by the setting. This is the peculiarity w.r.t. [18, 16]: in both papers, the proposed losses are defined using class-dependent weights (fixed to 1/2 for the first paper, and dependent on the class noise rate for the latter) while in our approach the used weights are provided for each instance, which gives a more flexible formulation. Making use of this β-risk , we design a generic algorithm devoted to address different kinds of aforementioned weakly labeled settings. To allow a comparison with the state of the art, we instantiate it with a learner that takes the form of an SVM algorithm. In this context, we derive a soft margin β-SVM algorithm and show that it outperforms WELLSVM.

The remainder of this paper is organized as follows: in Section 2, we define the empirical surrogate β-risk and show under which conditions it can be used to learn without explicitly accessing the labels; we also show how to instantiate β according to the weakly labeled learning setting at hand; in Section 3, we present our generic iterative algorithm for learning with weakly labeled data and in Section 4 we exploit our new framework to derive a novel formulation of the Support Vector Machine problem, the β-SVM ; finally, we report experiments in semi-supervised learning and learning with label noise, conducted on classical datasets from the UCI repository [15], in order to compare our algorithm with the state of the art approaches.

2 From Classical Surrogate Losses and Surrogate Risks to the β-risk

In this section, we first provide reminders about surrogate losses and then exploit the characteristics of the popular loss functions to introduce the empirical surrogate β-risk . The β-risk formulation allows us to tackle the problem of learning with weakly labeled data. We show under which conditions it can be used instead of the standard empirical surrogate risk (defined in a fully supervised context). Those conditions give insight on how to design algorithms that learn from weak supervision. We restrain our study to the context of binary classification.

2.1 Preliminaries

In statistical learning, a common approach for choosing the optimal hypothesis h^* from a hypothesis class $\mathcal{H}$ is to select the classifier that minimizes the expected risk over the joint space $Z = X \times Y$, where X is the feature space and Y the label space, expressed as

$$\mathcal{R}_\ell(h) = \int_{X \times Y} \ell(yh(x))p(x, y)dxdy$$

with $\ell : \mathcal{H} \times Z \to \mathbb{R}^+$ a margin-based loss function.

Since the true distribution of the data $p(x, y)$ is usually unknown, machine learning algorithms typically minimize the empirical version of the risk, computed over a finite set $\mathcal{S}$ composed of m instances (x_i, y_i) i.i.d. drawn from a distribution over $X \times \{-1, 1\}$:

$$\mathcal{R}_\ell(\mathcal{S}, h) = \frac{1}{m} \sum_{i=1}^{m} \ell(y_i h(x_i)).$$

The most natural loss function is the so-called 0-1 loss. As this function is not convex, not differentiable and has zero gradient, other loss functions are commonly employed instead. These losses, such as the logistic loss (e.g., for the logistic regression [6]), the exponential loss (e.g., for boosting techniques [10]) and the hinge loss (e.g., for the SVM [7]), are convex and smooth relaxations of the 0-1 loss. Theoretical studies on the characteristics and behavior of such *surrogate losses* can be found in [17, 2, 20]. In particular, [17] showed that each commonly used surrogate loss can be

characterized by a permissible function ϕ (see below) and rewritten as $F_\phi(x)$

$$F_\phi(x) = \frac{\phi^*(-x) - a_\phi}{b_\phi}$$

where $\phi^*(x) = \sup_a(xa - \phi(a))$ is the Legendre conjugate of ϕ (for more details, see [4]), $a_\phi = -\phi(0) = -\phi(1) \geq 0$ and $b_\phi = -\phi(\frac{1}{2}) - a_\phi > 0$. As presented by the authors of [13] and [17], a permissible function is a function $f : [0,1] \to \mathbb{R}^-$, symmetric about $-\frac{1}{2}$, differentiable on $]0,1[$ and strictly convex. For instance, the permissible function ϕ_{log} related to the logistic loss $F_\phi(x) = \log(1 + \exp^{-x})$ is:

$$\phi_{log}(x) = x\log(x) + (1-x)\log(1-x)$$

and $a_\phi = 0$ and $b_\phi = \log(2)$.

As detailed in [17], considering a surrogate loss F_ϕ, the empirical surrogate risk of an hypothesis $h : X \to \mathbb{R}$ w.r.t. $\mathcal{S}$ can be expressed as:

$$\mathcal{R}_\phi(\mathcal{S}, h) = \frac{1}{m}\sum_{i=1}^{m} D_\phi\big(y_i, \nabla_\phi^{-1}(h(x_i))\big) = \frac{b_\phi}{m}\sum_{i=1}^{m} F_\phi(y_i h(x_i))$$

with D_ϕ the Bregman Divergence

$$D_\phi(x,y) = \phi(x) - \phi(y) - (x-y)\nabla_\phi(y).$$

In order to evaluate such risk $\mathcal{R}_\phi(\mathcal{S}, h)$, it is mandatory to provide the labels y for all the instances. In addition, it is not possible to take into account eventual uncertainties on the given labels. Consequently, $\mathcal{R}_\phi$ is defined in a totally supervised context, where the labels y are known and considered to be true. In order to face the numerous situations where training data may be weakly labeled, we claim that there is a need to fill the gap by defining a new empirical surrogate risk that can deal with such settings. In the following section, we propose a generalization of the empirical surrogate risk, called the empirical surrogate $\beta-$risk, which can be employed in the context of weakly labeled data instead of the standard one under some linear conditions on the margin.

2.2 The Empirical Surrogate β-risk

Before defining the empirical surrogate β-risk for any loss F_ϕ and hypothesis $h \in \mathcal{H}$, let us rewrite the definition of $\mathcal{R}_\phi$ introducing a new set of variables named β, and that can be laid out as a $2\times m$ matrix.

Lemma 2.1. *For any $\mathcal{S}$, ϕ and h, and for any non-negative real coefficients β_i^{-1} and β_i^{+1} defined for each instance $x_i \in \mathcal{S}$ such that $\beta_i^{-1} + \beta_i^{+1} = 1$, the empirical surrogate risk $\mathcal{R}_\phi(\mathcal{S}, h)$ can be rewritten as*

$$\mathcal{R}_\phi(\mathcal{S}, h) = \mathcal{R}_\phi(\mathcal{S}, h, \beta)$$

where

$$\mathcal{R}_\phi(\mathcal{S}, h, \beta) = \frac{b_\phi}{m}\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{-1,+1\}}} \beta_i^\sigma F_\phi(\sigma h(x_i)) + \frac{1}{m}\sum_{i=1}^{m}\beta_i^{-y_i}(-y_i h(x_i)).$$

The coefficient β_i^{+1} (resp. β_i^{-1}) for an instance x_i can be interpreted here as the degree of confidence in (or the probability of) the label +1 (resp. -1) assigned to x_i.

Proof.

$$\begin{aligned}
\mathcal{R}_\phi(\mathcal{S}, h) &= \frac{b_\phi}{m} \sum_{i=1}^{m} F_\phi(y_i h(x_i)) \\
&= \frac{b_\phi}{m} \sum_{i=1}^{m} \left(\beta_i^{y_i} F_\phi(y_i h(x_i)) + \beta_i^{\text{-}y_i} F_\phi(y_i h(x_i)) \right) \quad (1) \\
&= \frac{b_\phi}{m} \sum_{i=1}^{m} \left(\beta_i^{y_i} F_\phi(y_i h(x_i)) + \beta_i^{\text{-}y_i} \left(F_\phi(-y_i h(x_i)) - \frac{y_i h(x_i)}{b_\phi} \right) \right) \quad (2) \\
&= \frac{b_\phi}{m} \sum_{i=1}^{m} \sum_{\substack{\sigma \in \\ \{\text{-}1,+1\}}} \beta_i^{\sigma} F_\phi(\sigma h(x_i)) + \frac{1}{m} \sum_{i=1}^{m} \beta_i^{\text{-}y_i} (-y_i h(x_i)). \quad (3)
\end{aligned}$$

Eq. (1) is because $\beta_i^{\text{-}1} + \beta_i^{+1} = 1$; Eq. (2) is due to the fact that $\phi^*(-x) = \phi^*(x) - x$ (see the supplementary material) for any permissible function ϕ, so that $F_\phi(x) = \frac{\phi^*(-x) - a_\phi}{b_\phi} = \frac{\phi^*(x) - a_\phi - x}{b_\phi} = F_\phi(-x) - \frac{x}{b_\phi}$. □

From Eq. (3), and considering that the sample $\mathcal{S}$ is composed by the finite set of features $\mathcal{X}$ and labels $\mathcal{Y}$, we can write that

$$\mathcal{R}_\phi(\mathcal{S}, h) = \mathcal{R}_\phi(\mathcal{S}, h, \beta) = \mathcal{R}_\phi^\beta(\mathcal{X}, h) - \frac{1}{m} \sum_{i=1}^{m} \beta_i^{\text{-}y_i} y_i h(x_i) \quad (4)$$

where

$$\mathcal{R}_\phi^\beta(\mathcal{X}, h) = \frac{b_\phi}{m} \sum_{i=1}^{m} \sum_{\substack{\sigma \in \\ \{\text{-}1,+1\}}} \beta_i^{\sigma} F_\phi(\sigma h(x_i))$$

is the empirical surrogate β-risk for a matrix $\beta = [\beta_0^{+1}, ..., \beta_m^{+1} | \beta_0^{\text{-}1}, ..., \beta_m^{\text{-}1}]$.

It is worth noticing that $\mathcal{R}_\phi(\mathcal{S}, h, \beta)$ is expressed in the form of a sum of two terms: the second one takes into account the labels of the data, while the first one, the β-risk, focuses on the loss suffered by h over $\mathcal{X}$ without explicitly needing the labels $\mathcal{Y}$.

The empirical β-risk is a generalization of the empirical risk: setting $\beta_i^{y_i} = 1$ (and thus $\beta_i^{-y_i} = 0$) for each instance, the second term vanishes and we retrieve the classical formulation of the empirical risk. Additionally, as developed in Section 2.3, the introduction of β makes it possible to inject some side-information about the labels. For this reason, we claim that the β-risk is suited to deal with classification in the context of weakly labeled data.

Let us now focus on the conditions allowing the empirical β-risk (i) to be a surrogate of the 0-1 loss-based empirical risk and (ii) to be sufficient to learn with a weak supervision on the labels. From (4), we deduce:

$$\mathcal{R}_\phi^\beta(\mathcal{X}, h) = \mathcal{R}_\phi(\mathcal{S}, h, \beta) + \frac{1}{m} \sum_{i=1}^{m} \beta_i^{\text{-}y_i} y_i h(x_i) \geq \mathcal{R}_{0/1}(\mathcal{S}, h) + \frac{1}{m} \sum_{i=1}^{m} \beta_i^{\text{-}y_i} y_i h(x_i) \quad (5)$$

where $\mathcal{R}_{0/1}(\mathcal{S}, h)$ the empirical risk related to the 0-1 loss and Eq. (5) is because $b_\phi F_\phi(x) \geq F_{0/1}(x)$ (for any surrogate loss).

It is possible to ensure that the β-risk is both a convex upper-bound of the 0-1 loss based risk and a relaxation as tight as the traditional risk (i.e., that we have $\mathcal{R}_{0/1}(\mathcal{S}, h) \leq \mathcal{R}_\phi^\beta(\mathcal{X}, h) \leq \mathcal{R}_\phi(\mathcal{S}, h)$) is to force the following constraint: $\sum_{i=1}^{m} \beta_i^{\text{-}y_i} y_i h(x_i) = 0$.

Unfortunately, the constraint $\sum_{i=1}^{m} \beta_i^{\text{-}y_i} y_i h(x_i) = 0$ still depends on the vector y of labels, which is not always provided and most likely uncertain or inaccurate in a weakly labeled data setting. We will show in Section 3 that this issue can be overcome by means of an iterative 2-step learning procedure, that first learns a classifier minimizing the β-risk , possibly violating the constraint, and then learns a new matrix β that fulfills the constraint.

2.3 Instantiating β for Different Weakly Supervised Settings

The β-risk can be used as the basis for handling different learning settings, including weakly labeled learning. This can be achieved by fixing the β values, choosing their initial values or putting a prior on them. We have already seen that, fully supervised learning can be obtained by fixing all β values to 1 for the assigned class and to 0 for the opposite class. The current section provides guidance on how β could be instantiated to handle various weakly labeled settings.

In a *semi-supervised* setting, as detailed in the experimental section, we propose to initialize the β of unlabeled points to 0.5 and then to automatically refine them in an iterative process. Going further, and if we are ready to integrate spatial or topological information in the process, the β values of each unlabeled point could be initialized using a density estimation procedure (e.g., by considering the label proportions of the k nearest labeled neighbors). In the context of *multi-expert learning*, the experts' votes for each instance i can simply be averaged to produce the β_i values (or their initialization, or a prior). The case of *learning with label proportions* is especially useful for privacy-preserving data processing: the training points are grouped into bags and, for each bag, the proportion of labels are given. One way of handling such supervision is to initialize, for each bag, all the β with the same value that corresponds to the provided proportion of labels. *Noise-tolerant learning* aims at learning in the presence of label noise, where labels are given but can be wrong. For any point that can be possibly noisy, a direct approach is to use lower β values (instead of 1 in the supervised case) and refine them as in the semi-supervised setting. β can also be initialized using the label proportion of the k nearest labeled example (as done in the experimental section). The case of *Multiple Instance Learning* (MIL) is trickier: in a typical MIL setting, instances are grouped in bags and the supervision is given as a single label per bag that is positive if the bag contains at least one positive instance (negative bags contain only negative instances). A straightforward solution would be to recast the MIL supervision as a "learning with label proportion" (e.g., considering exactly one positive instance in each bag). It is not fully satisfying and a more promising solution would be to consider, within each bag, the set of β^{+1} variables and put a sparsity-inducing prior on them. This approach would be a less-constrained version of the relaxation proposed in WellSVM [14] (where it is supposed that there is exactly one positive instance per positive bag) and could be achieved by a l_1 penalty or using a Dirichlet prior (with low α to promote sparsity).

3 An Iterative Algorithm for Weakly-labeled Learning

As explained in Section 2, a sufficient condition for guaranteeing that the β-risk is a convex upper-bound of the 0-1 loss based risk and it is not worse than the traditional risk is to fix $\sum_{i=1}^{m} \beta_i^{-y_i} y_i h(x_i) = 0$. However, the previous constraint depends on the labels. We overcome this problem by (i) iteratively learning a classifier minimizing the β-risk and most likely violating the constraint and then (ii) learning a new matrix β that fulfills it. The algorithm is generic. It can be used in different weakly labeled settings and can be instantiated with different losses and regularizations, as we will do in the next Section with SVMs.

As the process is iterative, let ${}^t\beta$ be the estimation of β at iteration t. At each iteration, our algorithm consists in two steps. We first learn an hypothesis h for the following problem P_1:

$$h^{t+1} = P_1(\mathcal{X}, {}^t\beta) = \arg\min_h \; c\mathcal{R}_\phi^{{}^t\beta}(\mathcal{X}, h) + \mathcal{N}(h)$$

which boils down to minimizing the $\mathcal{N}$-regularized empirical surrogate β-risk over the training sample $\mathcal{X}$ of size m, where $\mathcal{N}$, for instance, can take the form of a L_1 or a L_2 norm.

Then, we find the optimal β of the following problem P_2 for the points of $\mathcal{X}$:

$$^{t+1}\beta = P_2(\mathcal{X}, h^{t+1}) = \arg\min_\beta \mathcal{R}_\phi^\beta(\mathcal{X}, h^{t+1})$$

$$s.t. \sum_{i=1}^{m} \beta_i^{-y_i}(-y_i\, h^{t+1}(x_i)) = 0$$

$$\beta_i^{-1} + \beta_i^{+1} = 1, \;\; \beta_i^{-1} \geq 0, \beta_i^{+1} \geq 0 \;\; \forall i = 1..m\,.$$

For this step, a vector of labels is required. We choose to re-estimate it at each iteration according to the current value of β: we affect to an instance the most probable label, i.e. the σ corresponding

to the biggest β^σ. The matrix β has to be initialized at the beginning of the algorithm according to the problem setting (see Section 2.3). While some stabilization criterion does not exceed a given threshold ϵ, the two steps are repeated.

4 Soft-margin β-SVM

A major advantage of the empirical surrogate β-risk is that it can be plugged in numerous learning settings without radically modifying the original formulations. As an example, in this section we derive a new version of the Support Vector Machine problem, using the empirical surrogate β-risk , that takes into account the knowledge provided for each training instance (through the matrix β).

The soft-margin β-SVM optimization problem is a direct generalization of a standard soft-margin SVM and is defined as follows:

$$\arg\min_{\theta} \frac{1}{2}\|\theta\|_2^2 + c\sum_{i=1}^{m}\left(\beta_i^{-1}\xi_i^{-1} + \beta_i^{+1}\xi_i^{+1}\right)$$

$$s.t.\ \sigma(\theta^T\mu(x_i)+b) \geq 1-\xi_i^\sigma \ \ \forall i = 1..m, \sigma \in \{-1,1\}$$
$$\xi_i^\sigma \geq 0 \ \ \forall i = 1..m, \sigma \in \{-1,1\}$$

where $\theta \in X'$ is the vector defining the margin hyperplane and b its offset, $\mu : X \to X'$ a mapping function and $c \in \mathbb{R}$ a tuned hyper-parameter. In the rest of the paper, we will refer to $K : X \times X \to \mathbb{R}$ as the kernel function corresponding to μ, i.e. $K(x_i, x_j) = \mu(x_i)\mu(x_j)$.

The corresponding Lagrangian dual problem is given by (the complete derivation is provided in the supplementary material):

$$\max_{\alpha} \ -\frac{1}{2}\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{-1,+1\}}}\sum_{j=1}^{m}\sum_{\substack{\sigma'\in\\ \{-1,+1\}}}\alpha_i^\sigma\sigma\alpha_j^\sigma\sigma' K(x_i,x_j) + \sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{-1,+1\}}}\alpha_i^\sigma$$

$$s.t.\ 0 \leq \alpha_i^\sigma \leq c\beta_i^\sigma \ \ \forall i = 1..m,\ \sigma \in \{-1,1\}$$
$$\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{-1,+1\}}}\alpha_i^\sigma\sigma = 0 \ \ \forall i = 1..m,\ \sigma \in \{-1,1\}$$

which is concave w.r.t. α as for the standard SVM.

The β-SVM formulation differs from the SVM one in two points: first, the number of Lagrangian multipliers is doubled, because we consider both positive and negative labels for each instance; second, the upper-bounds for α are not the same for all instances but depend on the given matrix β. Like the coefficient c in the classical formulation of SVM, those upper-bounds play the role of trade-off between under-fitting and over-fitting: the smaller they are, the more robust to outliers the learner is but the less it adapts to the data. It is then logical that the upper-bound for an instance i depends on β_i^σ because it reflects the reliability on the label σ for that instance: if the label σ is unlikely, its corresponding α_i^σ will be constrained to be null (and its adversary will have more chance to be selected as a support vector, as $\beta_i^\sigma + \beta_i^{-\sigma} = 1$). Also, those points for which no label is more probable than the other ($\beta_i^\sigma \to 0.5$) will have less importance in the learning process compared to those for which a label is almost certain. In order to fully exploit the advantages of our formulation, c has to be finite and bigger than 0. As a matter of fact, when $c \to \infty$ or $c \to 0$, the constraints become exactly those of the original formulation.

5 Experimental Results

In the first part of this section, we present some experimental results obtained by adapting the iterative algorithm presented in Section 3 for semi-supervised learning and combining it with the previously derived β-SVM . Note that some approaches based on SVMs have been already presented in the literature to address the problem of semi-supervised learning. Among them, TransductiveSVM [5]

iteratively learns a separator with the labeled instances, classifies a subset of the unlabeled instances and adds it to the training set. On the other hand, WellSVM [14] combines the classical SVM with a label generation strategy that allows one to learn the optimal separator, even when the training sample is not completely labeled, by convexly relaxing the original Mixed-Integer Programming problem. In [14], WellSVM has been shown to be very effective and better than TransductiveSVM and the state of the art. For this reason, we compare in this section β-SVM to WellSVM. In the second subsection, we present some preliminary results in the noise-tolerant learning setting, showing how β-SVM behaves when facing data with label noise.

5.1 Iterative β-SVM for semi-supervised learning

We compare our method's performances to those of WellSVM, that has been proved, in [14], to performs in average better than the state of the art semi-supervised learning methods based on SVM and the standard SVM as well. In a semi-supervised context, a set $\mathcal{X}_l$ of labeled instances of size m_l and a set $\mathcal{X}_u$ of unlabeled instances of size m_u are provided. The matrix β is initialized as follows:

$$\forall i = 1..m_l \text{ and } \forall \sigma \text{ in } \{-1, 1\},\ {}^0\beta_i^\sigma = 1 \text{ if } \sigma = y_i, 0 \text{ otherwise,}$$

$$\forall i = m_l{+}1..m_u \text{ and } \forall \sigma \text{ in } \{-1, 1\},\ {}^0\beta_i^\sigma = 0.5$$

and we learn an optimal separator:

$$h^{t+1} = P_1(\mathcal{X}_l \cup \mathcal{X}_u, {}^t\beta) = \arg\min_h\ c_1\mathcal{R}_\phi^{{}^t\beta}(\mathcal{X}_l, h) + c_2\mathcal{R}_\phi^{{}^t\beta}(\mathcal{X}_u, h) + \mathcal{N}(h).$$

Here c_1 and c_2 are balance constants between the labeled and unlabeled set: when the number of unlabeled instances become greater than the number of labeled instances, we need to reduce the importance of the unlabeled set in the learning procedure because there exists the risk that the labeled set will be ignored. We consider the provided labels to be correct, so we keep the corresponding ${}_l\beta$ fixed during the iterations of the algorithm and estimate ${}_u\beta$ by optimizing $P_2(\mathcal{X}_u, h^{t+1})$. The iterative algorithm with β-SVM is implemented in Python using Cvxopt (for optimizing β-SVM) and Cvxpy [2] with its Ecos solver [9].

For each dataset, we show in Figure 1 the accuracy of the two methods with an increasing proportion of labeled data. The different approaches are compared on the same kernel, either the linear or the gaussian, the one that gives higher overall accuracy. As a matter of fact, the choice of the kernel depends on the geometry of the data, not on the learning method.

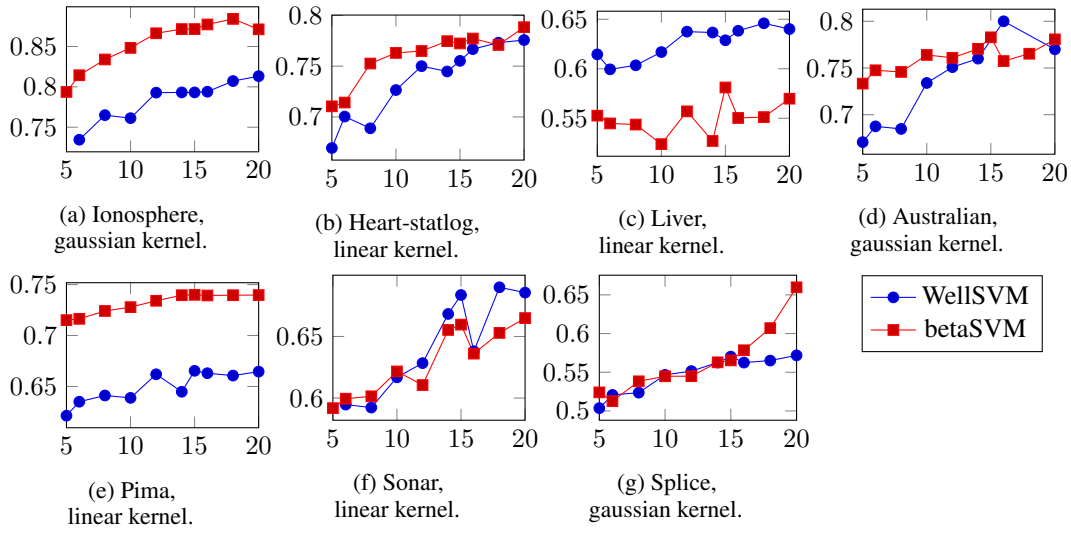
For each proportion of labeled data, we perform a 4-fold cross-validation and we show the average accuracy over 10 iterations. Concerning the hyper-parameters of the different methods, we fix c_2 of β-SVM to $c_1\frac{m_l}{m}$, c_1 of WellSVM to 1 as explained in [14] and all the other hyper-parameters (c_1 for β-SVM and c_2 for WellSVM) are tuned by cross-validation through grid search. As for the stopping criteria, we fix ϵ of β-SVM to $10^{-5} + 10^{-3}\|h\|_{\mathcal{F}}$ and ϵ of WellSVM to 10^{-3} and the maximal number of iterations to 20 for both methods. When using the gaussian kernel, the γ in $K(x_i, x_j) = \exp(-\|x_i - x_j\|_2^2/\gamma)$ is fixed to the mean distance between instances.

Our method performs better than WellSVM, with few exceptions, and is more efficient in terms of CPU time: for the Australian dataset, the biggest dataset in number of features and instances, WellSVM is in average 30 times slower than our algorithm (without particular optimization efforts).

5.2 Preliminary results under label-noise

We quickly tackle another setting of the weakly labeled data field: the noise-tolerant learning, the task of learning from data that have noisy or uncertain labels. It has been shown in [3] that SVM learning is extremely sensitive to outliers, especially the ones lying next to the boundary. We study, the sensitivity of β-SVM to label noise artificially introduced on the Ionosphere dataset. We consider two initialization strategies for β: the *standard* on where $\beta^{y_i} = 1$ and $\beta^{-y_i} = 0$ and the ***4-nn*** one where β^σ is set to the proportion of neighboring instances with label σ. In Figure 2, we draw the mean accuracy over 4 repetitions w.r.t. an increasing percentage (as a proportion of the smallest dataset) of two kinds of noise: the symmetric noise, introduced by swapping the labels of instances belonging to different classes, and the asymmetric noise, introduced by gradually changing the labels of the

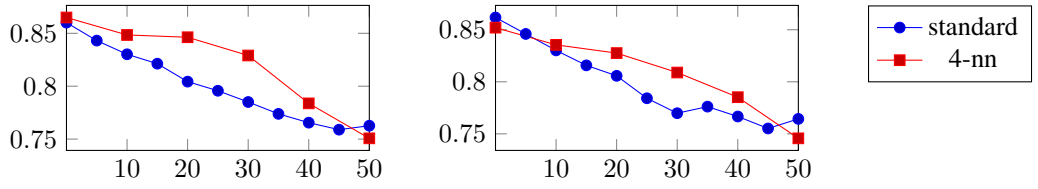
[2] http://cvxopt.org/ and http://www.cvxpy.org/



(a) Ionosphere, gaussian kernel. (b) Heart-statlog, linear kernel. (c) Liver, linear kernel. (d) Australian, gaussian kernel.

(e) Pima, linear kernel. (f) Sonar, linear kernel. (g) Splice, gaussian kernel.

Figure 1: Comparison of the mean accuracies of WellSVM and β-SVM versus the percentage of labeled data on different UCI datasets.



(a) Symmetric Noise. (b) Asymmetric Noise.

Figure 2: Comparison of the mean accuracy versus the percentage of noise of iterative β-SVM with different initializations of β. The *standard* curve refers to the initialization of $\beta^{y_i} = 1$ and $\beta^{-y_i} = 0$ and the *4-nn* to the initialization of β^{σ} to the proportion of neighboring instances with label σ.

instances of one class. These preliminary results are encouraging and show that locally estimating the conditional class density to initialize the β matrix improves the robustness of our method to label noise.

6 Conclusion

This paper focuses on the problem of learning from weakly labeled data. We introduced the β-risk which generalizes the standard empirical risk while allowing the integration of weak supervision. From the expression of the β-risk , we derived a generic algorithm for weakly labeled data and specialized it in an SVM-like context. The resulting β-SVM algorithm has been applied in two different weakly labeled settings, namely semi-supervised learning and learning with label noise, showing the advantages of the approach.

The perspectives of this work are numerous and of two main kinds: covering new weakly labeled settings and studying theoretical guarantees. As proposed in Section 2.3, the β-risk can be used in various weakly labeled scenarios. This requires to use different strategies for the initialization and the refinement of β, and also to propose proper priors for these parameters. Generalizing the proposed β-risk to a multi-class setting is a natural extension as β is already a matrix of class probabilities. Another broad direction involves deriving robustness and convergence bounds for the algorithms built on the β-risk .

7 Acknowledgments

We thank the reviewers for their valuable remarks. We also thank the ANR projects SOLSTICE (ANR-13-BS02-01) and LIVES (ANR-15-CE230026-03).

References

[1] D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

[2] S. Ben-David, D. Loker, N. Srebro, and K. Sridharan. Minimizing the misclassification error rate using a surrogate convex loss. In *Proceedings of the 29th International Conference on Machine Learning, ICML*. icml.cc / Omnipress, 2012.

[3] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[4] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[5] L. Bruzzone, M. Chi, and M. Marconcini. A novel transductive svm for semisupervised classification of remote-sensing images. *Geoscience and Remote Sensing, IEEE Transactions on*, 44(11):3363–3373, 2006.

[6] M. Collins, R. E. Schapire, and Y. Singer. Logistic regression, adaboost and bregman distances. *Machine Learning*, 48(1-3):253–285, 2002.

[7] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[8] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1):31–71, 1997.

[9] A. Domahidi, E. Chu, and S. Boyd. Ecos: An socp solver for embedded systems. In *Control Conference (ECC), 2013 European*, pages 3071–3076. IEEE, 2013.

[10] Y. Freund, R. E. Schapire, et al. Experiments with a new boosting algorithm. In *ICML*, volume 96, pages 148–156, 1996.

[11] T. Hastie, R. Tibshirani, and J. Friedman. *Unsupervised learning*. Springer, 2009.

[12] A. Joulin and F. Bach. A convex relaxation for weakly supervised classifiers. *arXiv preprint arXiv:1206.6413*, 2012.

[13] M. Kearns and Y. Mansour. On the boosting ability of top-down decision tree learning algorithms. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 459–468. ACM, 1996.

[14] Y.-F. Li, I. W. Tsang, J. T. Kwok, and Z.-H. Zhou. Convex and scalable weakly labeled svms. *The Journal of Machine Learning Research*, 14(1):2151–2188, 2013.

[15] M. Lichman. UCI machine learning repository, 2013.

[16] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204, 2013.

[17] R. Nock and F. Nielsen. Bregman divergences and surrogates for learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2048–2059, 2009.

[18] G. Patrini, F. Nielsen, R. Nock, and M. Carioni. Loss factorization, weakly supervised learning and label noise robustness. *arXiv preprint arXiv:1602.02450*, 2016.

[19] G. Patrini, R. Nock, T. Caetano, and P. Rivera. (almost) no label no cry. In *Advances in Neural Information Processing Systems*, pages 190–198, 2014.

[20] L. Rosasco, E. De Vito, A. Caponnetto, M. Piana, and A. Verri. Are loss functions all the same? *Neural Computation*, 16(5):1063–1076, 2004.

[21] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622. ACM, 2008.

[22] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

β-risk: a New Surrogate Risk for Learning from Weakly Labeled Data

Supplementary Material

Valentina Zantedeschi **Rémi Emonet**

Marc Sebban

firstname.lastname@univ-st-etienne.fr
Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School,
Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France

1 Overview

This supplementary material is organized as follows: in Section 2, we prove the property of a Legendre conjugate of a permissible function used in Eq.(2) of Sec.(2) of the paper; in Section 3, we derive the dual problem of a soft-margin β-SVM;

2 Legendre Conjugate of Permissible Functions

The Legendre conjugate of a differentiable and strictly convex function ϕ can be written as:

$$\phi^*(x) = x\nabla_\phi^{-1}(x) - \phi(\nabla_\phi^{-1}(x)).$$

In the case of a permissible function ϕ, its Legendre conjugate has the following property: $\phi^*(-x) = \phi^*(x) - x$.

Proof.

$$\begin{aligned}
\phi^*(x) &= -x\nabla_\phi^{-1}(-x) - \phi(\nabla_\phi^{-1}(-x)) \\
&= -x(1 - \nabla_\phi^{-1}(x)) - \phi(1 - \nabla_\phi^{-1}(x)) && (1) \\
&= -x + x\nabla_\phi^{-1}(x) - \phi(\nabla_\phi^{-1}(x)) && (2) \\
&= \phi^*(x) - x
\end{aligned}$$

Because of the symmetry of ϕ about $-\frac{1}{2}$, in Eq. (1) $\nabla_\phi^{-1}(-x) = 1 - \nabla_\phi^{-1}(x)$ and in Eq. (2) $\phi(1-x) = \phi(x)$. □

3 Derivation of Soft-margin β-SVM

The soft-margin β-SVM optimization problem is a direct generalization of a standard soft-margin SVM and is defined as follows:

$$\arg\min_\theta \frac{1}{2}\|\theta\|_2^2 + c\sum_{i=1}^{m}\left(\beta_i^{\text{-}1}\xi_i^{\text{-}1} + \beta_i^{+1}\xi_i^{+1}\right)$$

$$s.t.\ \sigma(\theta^T\mu(x_i) + b) \geq 1 - \xi_i^\sigma \ \ \forall i = 1..m, \sigma \in \{-1, 1\}$$

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

$$\xi_i^\sigma \geq 0 \ \forall i = 1..m, \sigma \in \{-1, 1\}$$

where $\theta \in X'$ is the vector defining the margin hyperplane and b its offset, $\mu : X \to X'$ a mapping function and $c \in \mathbb{R}$ a tuned hyper-parameter. In the rest of the paper, we will refer to $K : X^2 \to \mathbb{R}$ as the kernel function corresponding to μ ($K(x_i, x_j) = \mu(x_i)\mu(x_j)$).

Instead of solving the previous primal problem, it is preferable to solve its Lagrangian dual problem given by maximizing the corresponding Lagrangian w.r.t. its Lagrangian multipliers, which gives a nice Quadratic Programming problem that can be solved by common optimization techniques. The Lagrangian can be written as follows:

$$\mathcal{L}(\theta, b, \xi, \alpha, r) = \frac{1}{2}\|\theta\|_2^2 + c\sum_{i=1}^{m}\left(\beta_i^{-1}\xi_i^{-1} + \beta_i^{+1}\xi_i^{+1}\right) - \sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{-1,+1\}}}\alpha_i^\sigma\left(\sigma(\theta^T\mu(x_i)+b) + \xi_i^\sigma - 1\right) - \sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{-1,+1\}}} r_i^\sigma\xi_i^\sigma$$

where $\alpha \in \mathbb{R}^{2*m}$ and $r \in \mathbb{R}^{2*m}$ are the Lagrangian multipliers. It is obvious that:

$$\max_{\alpha,r\geq 0}\min_{\theta,b,\xi}\mathcal{L}(\theta, b, \xi, \alpha, r) \leq \min_{\theta,b,\xi}\max_{\alpha,r\geq 0}\mathcal{L}(\theta, b, \xi, \alpha, r)$$

where the left term corresponds to the optimal value of the dual problem and the right one to the primal's one. The dual and the primal problems have the same value at optimality if the Karush-Kuhn-Tucker (KKT) conditions are not violated (see [1]). By setting the gradient of $\mathcal{L}$ w.r.t. θ, b and ξ to 0, we find the saddle point corresponding to the function minimum:

$$\nabla_\theta\mathcal{L}(\theta, b, \xi, \alpha, r) = \theta - \sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{-1,+1\}}}\alpha_i^\sigma\sigma\mu(x_i)$$

$$\nabla_b\mathcal{L}(\theta, b, \xi, \alpha, r) = -\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{-1,+1\}}}\alpha_i^\sigma\sigma$$

$$\nabla_{\xi_i^\sigma}\mathcal{L}(\theta, b, \xi, \alpha, r) = c\beta_i^\sigma - \alpha_i^\sigma - r_i^\sigma$$

which give

$$\theta = \sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{-1,+1\}}}\alpha_i^\sigma\sigma\mu(x_i) \tag{3}$$

$$\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{-1,+1\}}}\alpha_i^\sigma\sigma = 0 \tag{4}$$

$$\alpha_i^\sigma \leq c\beta_i^\sigma \tag{5}$$

We can now write the QP dual problem by replacing θ by its expression (3) and simplifying following (4) and (5):

$$\max_\alpha \ -\frac{1}{2}\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{-1,+1\}}}\alpha_i^\sigma\sigma\sum_{j=1}^{m}\sum_{\substack{\sigma\in\\ \{-1,+1\}}}\alpha_j^\sigma\sigma K(x_i, x_j) + \sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{-1,+1\}}}\alpha_i^\sigma$$

$$s.t. \ 0 \leq \alpha_i^\sigma \leq c\beta_i^\sigma \ \forall i = 1..m, \ \sigma \in \{-1, 1\}$$
$$\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{-1,+1\}}}\alpha_i^\sigma\sigma = 0 \ \forall i = 1..m, \ \sigma \in \{-1, 1\}$$

which is concave w.r.t. α.

Proof.

$$\mathcal{L}(\alpha) = \frac{1}{2}\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}\alpha_i^\sigma\sigma\mu(x_i)\sum_{j=1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}\alpha_j^\sigma\sigma\mu(x_j) + c\sum_{i=1}^{m}\left(\beta_i^{\text{-}1}\xi_i^{\text{-}1} + \beta_i^{+1}\xi_i^{+1}\right)$$

$$-\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}\alpha_i^\sigma\left(\sigma\left(\left(\sum_{j=1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}\alpha_j^\sigma\sigma\mu(x_j)\right)\mu(x_i)+b\right)+\xi_i^\sigma-1\right)-\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}r_i^\sigma\xi_i^\sigma \quad (6)$$

$$= -\frac{1}{2}\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}\alpha_i^\sigma\sigma\mu(x_i)\sum_{j=1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}\alpha_j^\sigma\sigma\mu(x_j)+\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}\alpha_i^\sigma$$

$$+\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}\left(c\beta_i^\sigma-\alpha_i^\sigma-r_i^\sigma\right)\xi_i^\sigma - b\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}\alpha_i^\sigma\sigma \quad (7)$$

$$= -\frac{1}{2}\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}\alpha_i^\sigma\sigma\sum_{j=1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}\alpha_j^\sigma\sigma K(x_i,x_j)+\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}\alpha_i^\sigma \quad (8)$$

In Eq. (7) the third and the fourth terms are null because of (4) and (5). □

We need the following two additional constraints in order to respect the KKT conditions which justify guarantee that the optimal value found by solving the dual problem corresponds to the optimal value of the primal:

$$\alpha_i^\sigma\left(\sigma(\theta^T+b)-1+\xi_i^\sigma\right)=0\;\forall\, i=1..m, \sigma\in\{-1,1\}$$

$$r_i^\sigma\xi_i^\sigma=0\;\forall\, i=1..m, \sigma\in\{-1,1\}$$

Once the Lagrangian dual problem solved, the characteristic vector θ and offset b of the optimal margin hyperplane can be retrieved by means of the support vectors machine, i.e. the instances whose corresponding α_i^σ are strictly greater than 0:

$$\theta=\sum_{i)1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}\alpha_i^\sigma\sigma\mu(x_i)$$

$$b=\theta\mu(x_k)-\sigma_k$$

and the new instances can be classified :

$$y(x)=sign(\sum_{i=1}^{m}\sum_{\substack{\sigma\in\\ \{\text{-1,+1}\}}}(\alpha_i^\sigma\sigma K(x_i,x))+b)$$

4 Additional Experiments

4.1 Semi-Supervised Learning

We report a table of mean accuracies with their relative errors of the performances of standard SVM, WellSVM and our method on 7 UCI datasets with 5%,10% and 15% of labeled instances of the training sets.

dataset	% labeled	SVM	WellSVM	betaSVM
ionosphere	5	0.74 ± 0.02	0.72 ± 0.04	**0.77±0.03**
	10	0.78 ± 0.03	0.79 ± 0.03	**0.80±0.04**
	15	0.81 ± 0.01	**0.82±0.02**	0.81 ± 0.02
sonar	5	0.58 ± 0.06	0.58 ± 0.03	**0.59±0.05**
	10	0.65 ± 0.04	0.64 ± 0.04	**0.66±0.05**
	15	0.65 ± 0.02	**0.67±0.02**	**0.67±0.02**
liver	5	0.59 ± 0.02	**0.61±0.04**	0.55 ± 0.04
	10	0.61 ± 0.04	**0.64±0.03**	0.58 ± 0.03
	15	**0.64±0.04**	**0.64±0.03**	0.58 ± 0.03
splice	5	**0.53±0.07**	0.50 ± 0.07	**0.53±0.06**
	10	**0.56±0.02**	0.55 ± 0.05	0.55 ± 0.07
	15	**0.60±0.03**	0.56 ± 0.05	0.56 ± 0.04
heart-statlog	5	0.64 ± 0.04	0.55 ± 0.03	**0.71±0.04**
	10	0.72 ± 0.03	0.62 ± 0.02	**0.76±0.03**
	15	0.73 ± 0.02	0.63 ± 0.03	**0.77±0.02**
australian	5	0.72 ± 0.05	0.64 ± 0.01	**0.73±0.06**
	10	**0.73±0.03**	0.72 ± 0.04	**0.73±0.04**
	15	**0.76±0.07**	0.75 ± 0.03	0.75 ± 0.04
pima	5	0.65 ± 0.01	0.62 ± 0.03	**0.71±0.01**
	10	0.69 ± 0.01	0.63 ± 0.03	**0.72±0.01**
	15	0.71 ± 0.01	0.64 ± 0.03	**0.72±0.01**

4.2 Robustness to Label Noise

Here we report the results of applying β-SVM to a synthetic dataset and study its robustness to artificially induced label noise.

The synthetic dataset consists in 40 instances of 2 balanced classes: the instances of each class are uniformly distributed around a center point so that they can be easily classified by a linear separator to which we will refer as the true separator.

In Fig. 1, we compare the linear classifiers learned at each iteration of our iterative, algorithm with β-SVM, with a standard linear SVM and with the true separator. We conducted the experiment as follows: we apply the two methods first on the original dataset, then on a dataset where we swapped the label of a random instance of each class and so on with an increasing number of swapped labels.

We notice that our method is more robust to label noise: even though at the first iteration, we learn the same separator as the standard linear SVM, through the following iterations the algorithm converges to a separator closer to the true separator.

References

[1] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Figure 1: Artificially induced label noise: the baseline, here, corresponds to the separator learned with a classical SVM. The first figure shows the learned separators with the original labels, and the other figures show the results for an increasing number of swapped labels going from left to right and from to bottom.