HANDLING UNCERTAINTIES IN SVM CLASSIFICATION

Émilie NIAF\textsuperscript{1,2}, Rémi FLAMARY\textsuperscript{3}, Carole LARTIZIEN\textsuperscript{2}, Stéphane CANU\textsuperscript{3}

\textsuperscript{1}INSERM U556, Lyon, 69424, France
\textsuperscript{2}CREATIS, UMR CNRS 5220; INSERM U1044; INSA-Lyon; UCBL, Villeurbanne, 69621, France
\textsuperscript{3}LITIS EA 4108, INSA-Universite de Rouen, Saint-Etienne-du-Rouvray, 76801, France

ABSTRACT

This paper addresses the pattern classification problem arising when available target data include some uncertainty information. Target data considered here is either qualitative (a class label) or quantitative (an estimation of the posterior probability). Our main contribution is a SVM inspired formulation allowing to take into account class probabilities have been proposed in the literature \cite{3, 4}. In our method aims a finding the separating hyperplane maximizing classification performances.

Index Terms— support vector machines, maximal margin algorithm, uncertain labels.

1. INTRODUCTION

In the mainstream supervised classification scheme, an expert is required for labelling a set of data used then as inputs for training the classifier. However, for even an expert, this labeling task is likely to be difficult in many applications. In the end the training data set may contain inaccurate classes for some examples, which leads to non robust classifiers\cite{1}. For instance, this is often the case in medical imaging where radiologists have to outline what they think are malignant tissues over medical images without access to the reference histopathologic information. We propose to deal with these uncertainties by introducing probabilistic labels in the learning dataset of input vectors \((x_i)_{i=1...m}\) the latter of which being

- class labels: \(l_i = y_i \in \{-1, +1\} \) for \(i = 1...n\) (in classification),
- real values: \(l_i = p_i \in [0, 1]\) for \(i = n + 1...m\) (in regression).

\(p_i\), associated to point \(x_i\) allows to consider uncertainties about point \(x_i\)’s class. We define it as the posterior probability for class 1.

\[ p_i = p(x_i) = P(Y_i = 1 \mid X_i = x_i). \]

We define the associated pattern recognition problem as

\[
\min_w \quad \frac{1}{2}\|w\|^2 \\
\text{subject to} \quad \begin{cases} y_i (w^\top x_i + b) \geq 1, & i = 1...n \\ z_i^- \leq w^\top x_i + b \leq z_i^+, & i = n + 1...m \end{cases}
\]

Where boundaries \(z_i^-, z_i^+\) directly depend on \(p_i\). This formulation consists in minimizing the complexity of the model while forcing good classification and good probability estimation (close to \(p_i\)). Obviously, if \(n = m\), we are brought back to the classical SVM problem formulation.

Following the idea of soft margin introduced in regular SVM to deal with the case of inseparable data, we introduce...
slack variables $\xi_i$. This measure the degree of misclassification of the datum $x_i$, thus relaxing hard constraints of the initial optimization problem which becomes

$$\min_{w, \xi, \xi^-} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i + \tilde{C} \sum_{i=n+1}^{m} (\xi_i^- + \xi_i^+) \quad (2)$$

subject to

$$\begin{align*}
y_i (w^T x_i + b) & \geq 1 - \xi_i, & i = 1 \ldots n \\
z_i^- - \xi_i^- & \leq w^T x_i + b \leq z_i^+ + \xi_i^+, & i = n+1 \ldots m \\
0 & \leq \xi_i, & i = 1 \ldots n \\
0 & \leq \xi_i^- \text{ and } 0 \leq \xi_i^+, & i = n+1 \ldots m
\end{align*}$$

Parameters $C$ and $\tilde{C}$ are predefined positive real numbers controlling the relative weighting of classification and regression performances.

Let $\varepsilon$ be the labelling precision and $\delta$ the confidence we have in the labelling. Let’s define $\eta = \varepsilon + \delta$. Then, the regression problem consists in finding optimal parameters $w$ and $b$ such that

$$\left| \frac{1}{1 + e^{-a(w^T x_i + b)}} - p_i \right| < \eta,$$

Thus constraining the probability prediction for point $x_i$ to remain around $\frac{1}{1 + e^{-a(w^T x_i + b)}}$ within distance $\eta$.

The boundaries (where $w^T x_i + b = \pm 1$), define parameter $a$ as:

$$a = \ln\left(\frac{1}{\eta} - 1\right)$$

Finally:

$$\max(0, p_i - \eta) \leq \frac{1}{1 + e^{-a(w^T x_i + b)}} < \min(p_i + \eta, 1),$$

where $z_i^- = -\frac{1}{a} \ln\left(\frac{1}{p_i - \eta} - 1\right)$ and $z_i^+ = -\frac{1}{a} \ln\left(\frac{1}{p_i + \eta} - 1\right)$.

### 3. DUAL FORMULATION

We can rewrite the problem in its dual form, introducing Lagrange multipliers. We are looking for a stationary point for the Lagrange function $L$ defined as

$$L(w, b, \xi, \xi^-, \xi^+, \alpha, \beta, \mu^+ , \mu^-, \gamma^+, \gamma^-) =$$

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i + \tilde{C} \sum_{i=n+1}^{m} (\xi_i^- + \xi_i^+)$$

$$- \sum_{i=1}^{n} \alpha_i (y_i (w^T x_i + b) - (1 - \xi_i)) - \sum_{i=1}^{m} \beta_i \xi_i$$

$$- \sum_{i=n+1}^{m} \mu^-_i ((w^T x_i + b) - (z_i^- - \xi_i^-)) - \sum_{i=n+1}^{m} \gamma^-_i \xi_i^-$$

$$- \sum_{i=n+1}^{m} \mu^+_i ((z_i^+ + \xi_i^+) - (w^T x_i + b)) - \sum_{i=n+1}^{m} \gamma^+_i \xi_i^+$$

with $\alpha \geq 0, \beta \geq 0, \mu^+ \geq 0, \mu^- \geq 0, \gamma^+ \geq 0$ and $\gamma^- \geq 0$.

Computing the derivatives of $L$ with respect to $w, b, \xi, \xi^-$ and $\xi^+$ leads to the following optimality conditions:

$$\begin{align*}
0 & \leq \alpha_i \leq C, & i = 1 \ldots n \\
0 & \leq \mu_i^+ \leq \tilde{C}, & i = n+1 \ldots m \\
0 & \leq \mu_i^- \leq \tilde{C}, & i = n+1 \ldots m \\
w & = \sum_{i=1}^{n} \alpha_i y_i x_i - \sum_{i=n+1}^{m} (\mu_i^+ - \mu_i^-) x_i \\
y^T \alpha & = \sum_{i=n+1}^{m} (\mu_i^+ - \mu_i^-)
\end{align*}$$

where $e_1 = \left[ \underbrace{1, \ldots, 1}_{n \text{ times}} \underbrace{0, \ldots, 0}_{(m-n) \text{ times}} \right]^T$ and $e_2 = \left[ \underbrace{0, \ldots, 0}_{n \text{ times}} \underbrace{1, \ldots, 1}_{(m-n) \text{ times}} \right]^T$.

Calculations simplifications then lead to $L(w, b, \xi, \xi^-, \xi^+, \alpha, \beta, \mu, \gamma^+, \gamma^-) = -\frac{1}{2} w^T w + \sum_{i=1}^{n} \alpha_i + \sum_{i=n+1}^{m} \mu_i^- z_i^- - \sum_{i=n+1}^{m} \mu_i^+ z_i^+$.

Finally, let $\Gamma = \left[ \alpha_1 \ldots \alpha_n \mu_{n+1}^+ \ldots \mu_m^+ \mu_{n+1}^- \ldots \mu_m^- \right]$ be a vector of dimension $2m - n$. Then

$$w^T w = \Gamma^T G \Gamma$$

where

$$G = \begin{pmatrix}
K_1 & - & K_2 & K_3 \\
- & K_1^T & - & K_3 \\
K_2^T & - & K_3 \\
\end{pmatrix}$$

with

$$K_1 = \frac{1}{n} \sum_{i,j=1}^{n} y_i y_j x^T_i x_j$$

$$K_2 = \frac{1}{n} \sum_{i=n+1}^{m} x^T_i x_j$$

$$K_3 = \frac{1}{(m-n)} \sum_{i=n+1}^{m} x^T_i x_j$$

The dual formulation becomes

$$\min_{\Gamma} \frac{1}{2} \Gamma^T G \Gamma - e^T \Gamma,$$

with $\tilde{\varepsilon} = \frac{1}{n} \underbrace{\sum_{i=1}^{n} z_i^-}_{\text{n times}} \ldots \underbrace{\sum_{i=n+1}^{m} z_i^-}_{\text{n-m times}} \underbrace{\sum_{i=n+1}^{m} z_i^+}_{\text{n times}} \ldots \underbrace{\sum_{i=m+1}^{m} z_i^+}_{\text{n-m times}}$.

and

$$0 \leq \Gamma \leq \left[ \underbrace{C \ldots C}_{\text{n times}} \underbrace{\tilde{C} \ldots \tilde{C}}_{\text{n times}} \right]^T$$

### 4. KERNELIZATION

Formulations (2) and (3) can be easily generalized by introducing kernel functions. Let $k$ be a positive kernel satisfying Mercer’s condition and $H$ the associated Reproducing Kernel Hilbert Space (RKHS). Within this framework equation (2) becomes

$$\min_{f, b, \xi, \xi^-} \frac{1}{2} \|f\|^2_H + C \sum_{i=1}^{n} \xi_i + \tilde{C} \sum_{i=n+1}^{m} (\xi_i^- + \xi_i^+) \quad (4)$$
subject to
\[
\begin{aligned}
&y_i(f(x_i) + b) \geq 1 - \xi_i, & i = 1 \ldots n \\
&z_i^+ - \xi_i^- \leq f(x_i) + b \leq z_i^+ + \xi_i^+, & i = n + 1 \ldots m \\
&0 \leq \xi_i, & i = 1 \ldots n \\
&0 \leq \xi_i^- \text{ and } 0 \leq \xi_i^+, & i = n + 1 \ldots m
\end{aligned}
\]

Formulation (3) remains identical, with
\[
\begin{align*}
K_1 &= (y_i y_j k(x_i, x_j))_{i,j=1 \ldots n}, \\
K_2 &= (k(x_i, x_j))_{i=1 \ldots n, j=n+1 \ldots m}, \\
K_3 &= (k(x_i, x_j))_{i,j=n+1 \ldots m},
\end{align*}
\]

5. EXAMPLES

In order to experimentally evaluate the proposed method for handling uncertain labels in SVM classification, we have simulated different data sets described below. In these numerical examples, a RBF kernel \(k(u, v) = e^{-\|u-v\|^2/2\sigma^2}\) is used and \(C = \hat{C} = 100\). We implemented our method using the SVM-KM Toolbox [8]. We compare the classification performances and probabilistic predictions of the C-SVM and P-SVM approaches. In the first case, probabilities are estimated by using Platt’s scaling algorithm [3] while in the second case, probabilities are directly estimated via the formula defined in (2):
\[
P(y=1|x) = \frac{1}{1+e^{-(a(w^T + b))}}.
\]
Performances are evaluated by computing

- Accuracy (Acc)
  Proportion of well predicted examples in the test set (for evaluating classification).

- Kullback Leibler distance (KL)
  \[
  D_{KL}(P||Q) = \sum_{i=1}^{n} P(y_i = 1|x_i) \log(\frac{P(y_i = 1|x_i)}{Q(y_i = 1|x_i)})
  \]
  for probability distributions P and Q (for evaluating probability estimation).

5.1. Probability estimation

We generate two unidimensional datasets, labelled ’+1’ and ’-1’, from normal distributions of variances \(\sigma_2^2 = \sigma_1^2 = 0.3\) and means \(\mu_1 = 0.5\) and \(\mu_2 = -0.5\). Let’s \((x_i')_{i=1 \ldots n'}\) denote the learning data set \((n'=200)\) and \((x_i')_{i=\ldots n'}\) the test set \((n'=1000)\). We compute, for each point \(x_i\), its true probability \(P(y_i = +1|x_i)\) to belong to class ’+1’. From here on, learning data are labelled in two ways, as follows

a) For \(i = 1 \ldots n\), we get the regular SVM dataset by simply using a probability of 0.5 as the threshold for assigning class labels \(y_i\) associated to point \(x_i\). This is what would be done in practical cases when the data contains class membership probabilities and a SVM classifier is used.

\[
\begin{align*}
\text{if } & P(y_i = 1|x_i') > 0.5, \text{ then } y_i' = 1, \\
\text{if } & P(y_i = 1|x_i') \leq 0.5, \text{ then } y_i' = -1
\end{align*}
\]

This dataset \((x_i', y_i')_{i=1 \ldots n}\) is used to train the C-SVM classifier.

b) We define another data set \((x_i', y_i')_{i=1 \ldots n'}\) such that, for \(i = 1 \ldots n\),

\[
\begin{align*}
\text{if } & P(y_i = 1|x_i') > 1 - \eta, \text{ then } y_i' = 1, \\
\text{if } & P(y_i = 1|x_i') < \eta, \text{ then } y_i' = -1, \\
& y_i' = P(y_i = 1|x_i') \text{ otherwise}
\end{align*}
\]

If the probability values are sufficiently close to 0 or 1 (closeness being defined by the precision and confidence), we admit that they belong respectively to class -1 or 1. This probabilistic dataset \((x_i', y_i')_{i=1 \ldots n'}\) is used to train the P-SVM algorithm.

We compare our two approaches using the test set \((x_i')_{i=1 \ldots n'}\).

As we know the true probabilities \(P(y_i = 1|x_i')\), we can estimate the probability prediction error (KL). Figure 1 shows the probability predictions improvement shown by the P-SVM: the true probabilities (black) and P-SVM estimations (red) are quasi-superimposed (KL=0.2) whereas Platt’s estimations are less accurate (KL=11.3).

\[
\text{Fig. 1: Probability estimations comparison. Top plot shows the true posterior probabilities with C-SVM and P-SVM estimations overlaying. Lower plot shows the distance between true probabilities and estimations.}
\]

5.2. Noise robustness

We generate two 2D datasets, labelled ’+1’ and ’-1’, from normal distributions of variances \(\sigma_2^2 = \sigma_1^2 = 0.7\) and means \(\mu_1 = (-0.3, -0.5)\) and \(\mu_2 = (+0.3, +0.5)\). As in the previous experiment, we compute class ’1’ membership probability for each point \(x_i\) of the learning data set. We simulate classification error by artificially adding a centered uniform noise (\(\delta\) of amplitude 0.1), to the probabilities, such that for \(i = 1 \ldots n\),

\[
\tilde{P}(y_i = 1|x_i) = P(y_i = 1|x_i) + \delta_i.
\]

We then label learning data following the same scheme as described in (5) and (6). Figure 2 shows the margin location and probabilities estimations using the two methods over a grid of values. Far from learning data points, both probability estimations are less accurate, this being directly linked to
the choice of a gaussian kernel. However, P-SVM classification and probability estimations obtained for 1000 test points, are clearly more alike the ground truth (Acc_{P-SVM} = 99%, KL_{P-SVM} = 3.6) than C-SVM (Acc_{C-SVM} = 95%, KL_{C-SVM} = 95). Contrary to P-SVM which, by combining both classification and regression, predicts good probabilities, C-SVM is sensitive to classification noise and is no more converging to the Bayes rule as seen in [1].

Contrary to P-SVM which, by combining both classification and regression, predicts good probabilities, C-SVM is sensitive to classification noise and is no more converging to the Bayes rule as seen in [1].

**Fig. 2:** Probability estimations of C-SVM and P-SVM over a grid using noisy learning data (uniform noise, amplitude 0.1). Noisy learning data are plotted in blue (class ‘-1’) and red (class ‘1’) stars.

Figure 3 shows the impact of noise amplitude on classifiers performances (values are averaged over 30 random simulations). Even if noise increases, classifications and probability predictions performances of the P-SVM remain significantly higher than those of C-SVM.

**6. CONCLUSION**

This paper has presented a new way to take into account both qualitative and quantitative target data by shrewdly combin-
$P(x|y=1)$

$P(x|y=-1)$

$P(y=1|x)$

$P(y=1|x)$ C−SVM + Platt

$P(y=1|x)$ P−SVM
C–SVM + Platt probability estimates
C−SVM + Platt probability estimates
P-SVM probability estimates
Platt probability estimates
outputs of the extended SVM
outputs of the extended SVM
