University of Illinois Fall 2018

ECE 544NA: Pattern Recognition

Lecture 8: September 25

Support Vector Machines (SVM)

Lecturer: Alexander Schwing Scribe: ANWESA CHOUDHURI

1 Introduction

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples [2]. The prediction only depends on a subset of the training data, known as the support vectors. The technique was originally designed by Boser, Guyon and Vapnik [1] for binary classification, but can be extended to multi-class classification and regression as well [3]. Logistic regression would be equally applicable for classification but SVMs were the dominant approach before 2010 because of higher empirical accuracy.

2 Binary SVM

We are given the training dataset $\mathcal{D} = \big\{(\phi(x^{(i)}, y^{(i)})\big\} \, \forall i \in \{1, 2, ..., N\}$, with the input data $x^{(i)} \in \mathbb{R}^d$, d being the dimension of input data and N being the number of training examples. The label corresponding to each is $x^{(i)}$ is $y^{(i)} \in \{-1, 1\}$. Multiple features for the training data are defined as $\phi(x^{(i)})$.

2.1 Intuition of Margin

In figure 1, we have 2 classes denoted by crosses and circles. The crosses are entirely above the line $w^T\phi = 1$ and circles are below $w^T\phi = 1$, w being the weight vector.

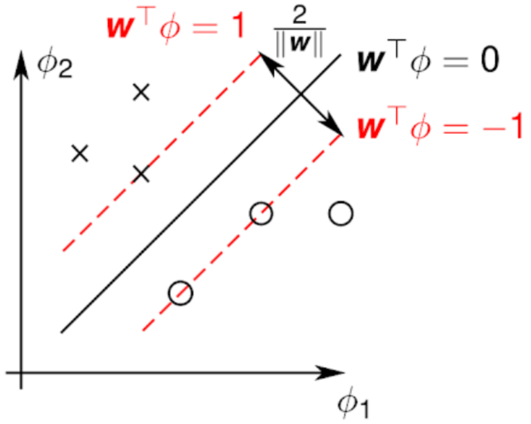


Figure 1: SVM Margin

Our aim is to maximize the margin, i.e., the distance between the two lines, $\frac{2}{\|w\|}$. This is formalized as follows [5].

$$\min_{w} \frac{C}{2} \parallel w \parallel_2^2 \qquad s.t, \qquad y^{(i)} w^T \phi(x^{(i)}) \geq L, \qquad \forall (\phi(x^{(i)}), y^{(i)}) \in \mathcal{D} \tag{1}$$

The term L is called the taskloss, $L \geq 0$.
C is the regularization parameter, that controls the degree of importance given to mis-classification. $L = 1$ and $C = 1$ in our example (figure 1).

2.2 Slack Variables

The above works for linearly separable data, but for non-linear cases, we introduce the notion of soft margins using slack variables. Using kernels is another technique which will not be discussed here. Slack variables ξ are introduced in equation 2 and figure 2 .

$$\min_{w, \xi^{(i)} \geq 0} \frac{C}{2} \parallel w \parallel_2^2 + \sum_{i \in \mathcal{D}} \xi^{(i)} \qquad s.t, \qquad y^{(i)} w^T \phi(x^{(i)}) \geq 1 - \xi^{(i)}, \qquad \forall (\phi(x^{(i)}), y^{(i)}) \in \mathcal{D} \tag{2}$$

Taskloss L is considered 1 here. Equivalently,

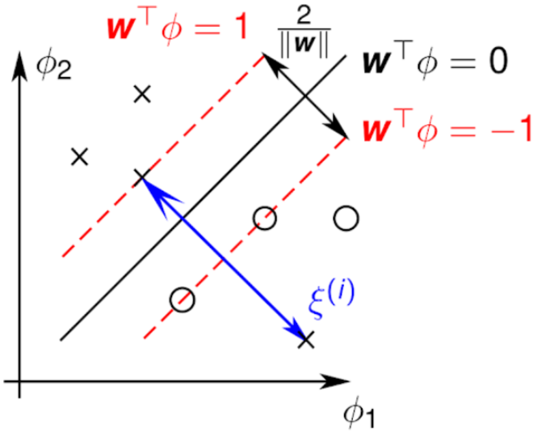


Figure 2: Slack Variables

$$\min_{w} \frac{C}{2} \parallel w \parallel_2^2 + \sum_{i \in \mathcal{D}} \max\{0, 1 - y^{(i)} w^T \phi(x^{(i)})\} \tag{3}$$

The first part of the equation is the quadratic loss and second part is the hinge loss. The hinge loss is the SVM's loss/error function of choice, whereas the l2-regularizer reflects the complexity of the solution, and penalizes complex solutions.
Emperical risk minimization can be written as follows:

$$\min_{w} R(w) + \sum_{i \in \mathcal{D}} \ell(y^{(i)}, F(x^{(i)}, w)) \tag{4}$$

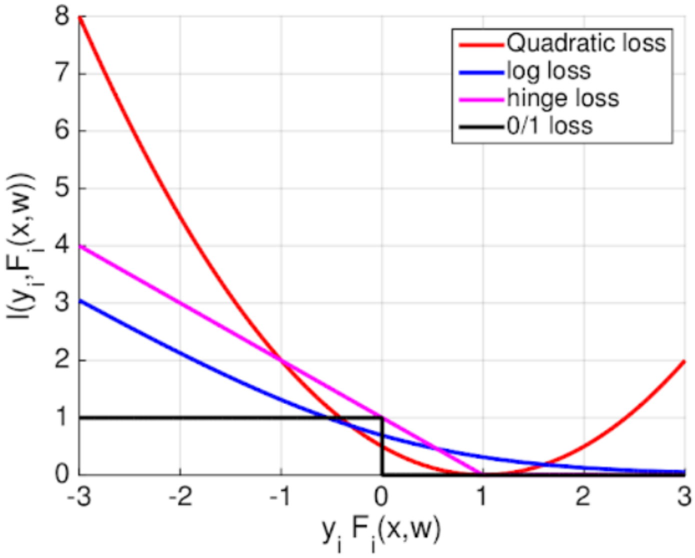


Figure 3: Comparison of quadratic loss, 0/1 loss, hinge loss and log loss

3 Optimization of binary SVM objective

3.1 Optimize Primal by Gradient Descent

The objective function for SVM is as shown below.

$$\min_{w} \frac{C}{2} \parallel w \parallel_2^2 + \sum_{i \in \mathcal{D}} max\{0, 1 - y^{(i)} w^T \phi(x^{(i)})\} \tag{5}$$

The gradient of $max\{0, x\}$: $\delta(x \geq 0)$ is 0 if $x < 0$ and 1 otherwise.
This is also useful for max-pooling.

3.2 Optimize Dual Problem

The primal objective with constraints is given as.

$$\min_{w, \xi^{(i)} \geq 0} \frac{C}{2} \parallel w \parallel_2^2 + \sum_{i \in \mathcal{D}} \xi^{(i)} \qquad s.t, \qquad y^{(i)} w^T \phi(x^{(i)}) \geq 1 - \xi^{(i)}, \qquad \forall (\phi(x^{(i)}), y^{(i)}) \in \mathcal{D} \tag{6}$$

The Lagrangian or the dual objective is obtained as shown below. The dual variables are $\alpha^{(i)} \geq 0$ for each inequality constraint.
Lagrangian :

$$\frac{C}{2} \parallel w \parallel_2^2 + \sum_{i \in \mathcal{D}} \xi^{(i)} + \sum_{i \in \mathcal{D}} \alpha^{(i)} (1 - \xi^{(i)} - y^{(i)} w^T \phi(x^{(i)}))$$
$$= \frac{C}{2} \parallel w \parallel_2^2 - w^T \sum_i \alpha^{(i)} y^{(i)} \phi(x^{(i)}) + \sum_i \xi^{(i)} (1 - \alpha^{(i)}) + \sum_i \alpha^{(i)}$$

To optimize the dual program, the primal program is first brought to the standard form, Lagrange multipliers are assigned to suitable set of constraints and the lagrangian is optimized with respect to the primal variables.

1. With respect to parameters w:

$$\frac{\delta L}{\delta W} : Cw = \sum_i \alpha^{(i)} y^{(i)} \phi(x^{(i)}) \tag{7}$$

2. With respect to slack variables:

$$\min_{\xi^{(i)} \geq 0} \xi^{(i)}(1 - \alpha^{(i)}) \Rightarrow \alpha^{(i)} \leq 1 \tag{8}$$

Dual program:

$$\max_{0 \leq \alpha \leq 1} g(\alpha) := \frac{-1}{2C} \parallel \sum_i \alpha^{(i)} y^{(i)} \phi(x^{(i)}) \parallel_2^2 + \sum_i \alpha^{(i)} \tag{9}$$

The dual is quadratic, hence quadratic program solvers are directly applicable to obtain the solution. One such technique is sequential minimal optimization [4].

3.3 Semi-definite Programming for SVM

Here we intend to find the solution by computing the dual of the dual. The equivalent dual program goes as follows.

$$\min_{0 \leq \alpha \leq 1} \hat{g}(\alpha) = \frac{1}{2C} \alpha^T A^T A \alpha - \sum_i \alpha^{(i)} \tag{10}$$

To obtain the dual of the dual, the Lagrangian is written using multipliers $\nu^{(i)} \geq 0, \gamma^{(i)} \geq 0$, where $\nu^{(i)}$ and $\gamma^{(i)}$ are the lagrange multipliers for the equality and inequality constraints.

$$\begin{aligned} L(.) &= \frac{1}{2C} \alpha^T A^T A \alpha - \sum_i \alpha^{(i)} - \sum_i \nu^{(i)} \alpha^{(i)} + \sum_i \gamma^{(i)} (\alpha^{(i)} - 1) \\ &= \frac{1}{2C} \alpha^T A^T A \alpha - \sum_i \alpha^{(i)} (1 + \nu^{(i)} - \gamma^{(i)}) - \sum_i \gamma^{(i)} \end{aligned} \tag{11}$$

The Lagrangian is optimized with respect to primal variables to obtain the dual program.
With respect to dual variables α:

$$\frac{\delta L}{\delta \alpha} : \frac{1}{C} A^T A \alpha = 1 + \nu + \gamma \tag{12}$$

Dual program of the dual goes as follows.

$$\begin{aligned} &\max_{\nu \geq 0, \gamma \geq 0} \frac{-1}{2C} (1 + \nu + \gamma)^T (A^T A)^{-1} (1 + \nu + \gamma) - \sum_i \gamma^{(i)} \\ \Rightarrow &\min_{\nu \geq 0, \gamma \geq 0} \frac{1}{2C} (1 + \nu - \gamma)^T (A^T A)^{-1} (1 + \nu - \gamma) + \sum_i \gamma^{(i)} \end{aligned} \tag{13}$$

Equivalent formulation would be:

$$\min_{\nu \geq 0, \gamma \geq 0} t \quad s.t \quad t \geq \frac{1}{2C} (1 + \nu - \gamma)^T (A^T A)^{-1} (1 + \nu - \gamma) + \sum_i \gamma^{(i)} \tag{14}$$

Lemma 1 *Schur Complement Lemma.*
Assuming $A^TA > 0$,

$$t - \frac{1}{2C}(1+\nu-\gamma)^T(A^TA)^{-1}(1+\nu-\gamma) - \sum_i \gamma^{(i)} \geq 0 \tag{15}$$

if and only if the following holds.

$$\begin{bmatrix} A^TA & (1+\nu-\gamma) \\ (1+\nu-\gamma)^T & C(t-\sum_i \gamma^{(i)}) \end{bmatrix} \geq 0 \tag{16}$$

There are different methods and use cases for those programs. Some of them are enumerated as follows.

- Interior point methods
- Chunking, sequential minimization optimization
- Coordinate ascent
- Active set methods
- Solving the primal with Newton's method
- Stochastic subgradient with projection
- Cutting plane algorithms

4 Relation between Logistic Regression and SVMs

Loss functions for linear and logistic regression and SVMs is given below.

- Linear Regression:
$$\min_w \frac{C}{2} \parallel w \parallel_2^2 + \sum_{i\in\mathcal{D}} \frac{1}{2}(1 - y^{(i)}w^T\phi(x^{(i)}))^2 \tag{17}$$
- Logistic Regression:
$$\min_w \frac{C}{2} \parallel w \parallel_2^2 + \sum_{i\in\mathcal{D}} log(1 + exp(-y^{(i)}w^T\phi(x^{(i)}))) \tag{18}$$
- Binary SVM:
$$\min_w \frac{C}{2} \parallel w \parallel_2^2 + \sum_{i\in\mathcal{D}} \max\left\{0, 1 - y^{(i)}w^T\phi(x^{(i)})\right\} \tag{19}$$

There are other kinds of loss functions like ramp loss, orbit loss, direct loss, etc.

4.1 Combining Log and Hinge Loss

We define F as a function of weights, input data and output label as follows.

$$F(x^{(i)}, w, y^{(i)}) = y^{(i)} w^T \phi(x^{(i)}) \tag{20}$$

Let $\epsilon > 0$ be a very small number. So, when $F \geq 0$, the following holds.

$$\lim_{\epsilon \to 0} log(1 + exp(\frac{-F}{\epsilon})) = 0 \tag{21}$$

Similarly when $F \leq 0$, we obtain L'Hospital's Rule to get the following.

$$\begin{aligned} \lim_{\epsilon \to 0} log(1 + exp(\frac{-F}{\epsilon}) &= \lim_{\epsilon \to 0} \frac{log(1 + exp(\frac{-F}{\epsilon}))}{1/\epsilon} \\ &= \lim_{\epsilon \to 0} \frac{\frac{exp\frac{-F}{\epsilon}}{1+exp\frac{-F}{\epsilon}}(F/\epsilon^2)}{-1/\epsilon^2} \\ &= \lim_{\epsilon \to 0} \frac{1}{1 + exp\frac{F}{\epsilon}}(-F) \\ &= -F \end{aligned} \tag{22}$$

Summarizing the above,

$$\lim_{\epsilon \to 0} log(1 + exp(\frac{-F}{\epsilon}) = \max\{0, F\} \tag{23}$$

This shows that SVM can be considered as 0-temperature limit of logistic regression.

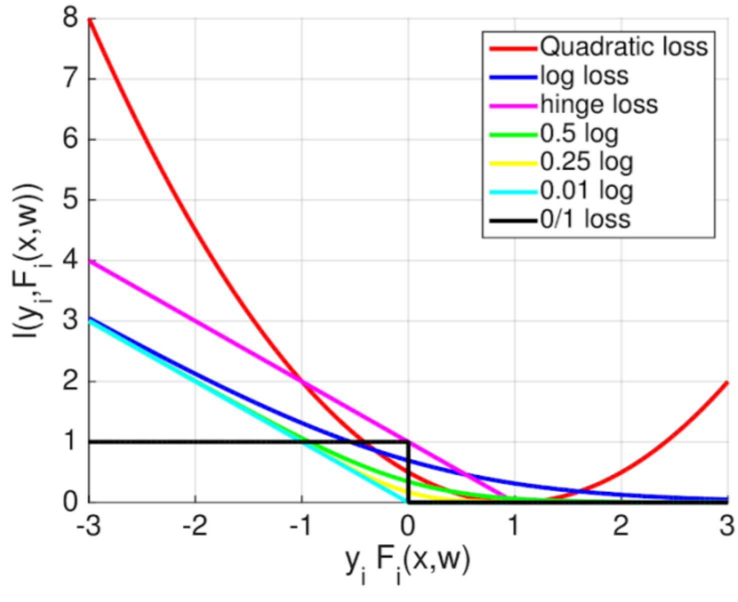


Figure 4: Hinge loss can be considered as a 0 log loss function. Log loss (purple) gradually takes shape of hinge loss as the value multiplied with it reduces(0.5 loss to 0.25 loss to 0.01 loss which looks very similar to hinge loss.)

The loss function for general binary classification can be written as the following.

- General binary classification

$$\min_{w} \frac{C}{2} \parallel w \parallel_2^2 + \sum_{i \in \mathcal{D}} \epsilon.log(1 + exp\frac{(L - y^{(i)} w^T \phi(x^{(i)})}{\epsilon}))) \quad (24)$$

5 An application of SVM: Object Detection

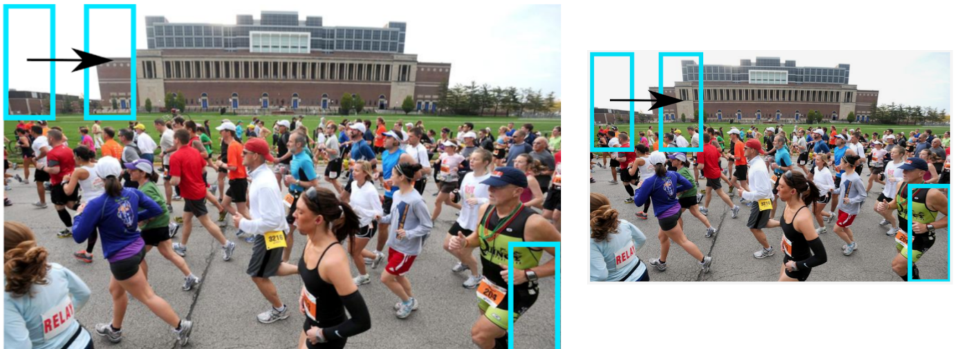


Figure 5: Bounding box of the same size is scanned at different scales and locations of the image

The following steps are followed for object detection.

- Figure 5 shows an RGB image. A bounding box (shown in blue) is scanned at different locations of the same image. This process is repeated for different scales of the same image.
- Multiple features are extracted from each bounding box. Some examples of good features are histogram of oriented gradients, as illustrated in figure 6.
- The SVM classifier is run on the various extracted features to find the object of interest (humans in this case).

Figure 7 shows the performance of linear or kernel SVMs over various features (HOG, SIFT, PCA, etc).

Figure 8 shows a successful result of human object detection with the numbers at the top left corner of the detected bounding box printing the confidence score.

References

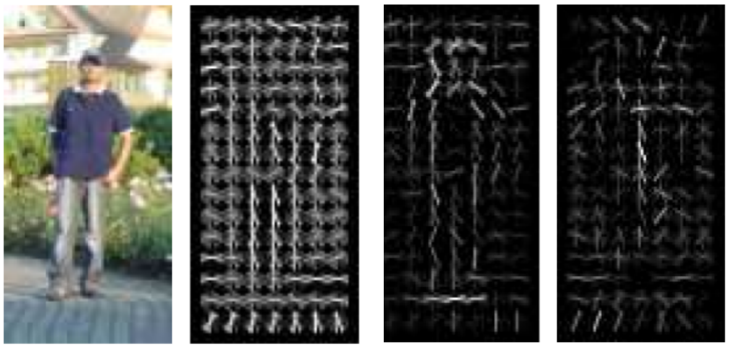
[1] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

[2] Itseez. *The OpenCV Reference Manual*, 2.4.9.0 edition, April 2014.

[3] K. P. Murphy. *Machine Learning: A Probabilistic Perspective.* The MIT Press, 2012.

[4] J. C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, Advances in kernel methods -Support Vector Learning, 1998.

[5] A. Schwing. Ece 544 pattern recognition: L8: Support vector machines. Technical report, UIUC, 2018.



(a) (b) (c) (d)

Figure 6: HOG features (b,c,d) for the RGB image (a)

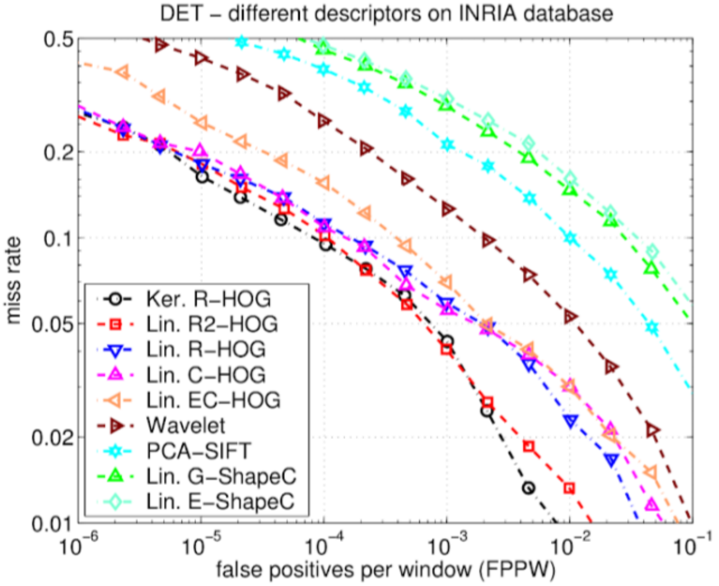


Figure 7: Comparison of Performance of Various kinds of features over linear and kernel SVMs

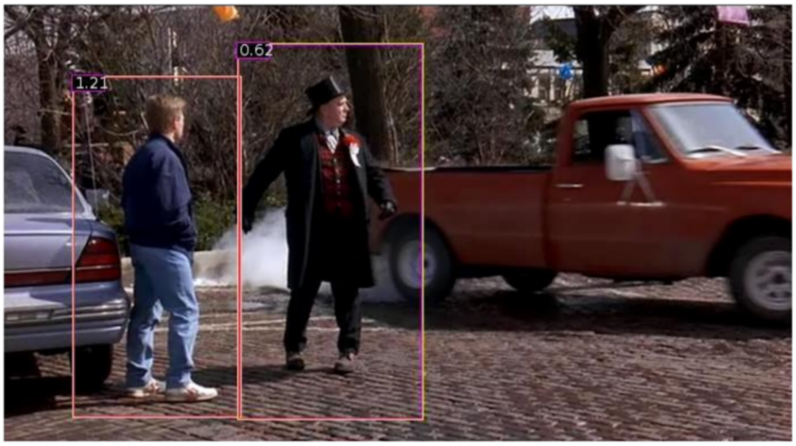


Figure 8: A successful detection