Support Vector Machines

Kartik Srinivas *ES20BTECH11015*
Tanmay Garg *CS20BTECH11063*
Aayush Patel *CS20BTECH11001*
Dishank Jain *AI20BTECH11011*
Gautham Bellamkonda *CS20BTECH11017*

***Abstract*—The support vector machine is a classification algorithm first proposed by Vladimir Vapnik in 1963 and further developed by Vapnik et al. during the 1990s. For this project, we have revisited the main ideas behind SVM's and programmed it form scratch to solve the task of spam classification.**

***Index Terms*—Support Vector Machines, Spam Classification, Structural risk minimisation, Kernel SVM**

I. Introduction

A classification problem can be loosely stated as:

Problem 1: Given a set of vectors $(\vec{x}_1, \vec{x}_2, ..., \vec{x}_n)$, and a set of labels $(y_1, y_2, ..., y_n)$, find a function f such that f($\vec{x}_i$) = y_i for all $i = 1, 2, ..., n$.

The SVM tries to solve this problem by assuming f to be a separating hyperplane. It operates on the binary classification problem class, i.e., each y_i can be either -1 or 1. More specifically, the SVM tries to find a plane: $\vec{w}^\top \vec{x} + b = 0$ such that

$$f(\vec{x}_i) = \begin{cases} 1 \; if \; \vec{w}^\top \vec{x}_i + b \geq 0 \\ -1 \; if \; \vec{w}^\top \vec{x}_i + b < 0 \end{cases} \tag{1}$$

We note that such a hyperplane may exist only if the vectors are linearly separable. Also, if the vectors are linearly separable, such a hyperplane may not be unique. Here, we require help from structural risk minimisation.

A set S of examples is shattered by a set of functions $\mathcal{H}$ if for every partition of the examples in S into positive and negative examples there is a function in the Hypothesis space $\mathcal{H}$ that gives exactly these labels to the examples. The VC-dimension of a hypothesis class $\mathcal{H}$, denoted $VCdim(\mathcal{H})$, is the maximal size of a set $C \subset X$ (X is the instance space) that can be shattered by $\mathcal{H}$.

Structural risk minimisation bounds the difference between **testing risk** and **training risk** with a certain **probability** $= 1 - \eta$ using the following equation

$$R^{test}(\lambda) \leq R^{train}(\lambda) + \sqrt{\frac{VCdim(\mathcal{H})(\ln \frac{2l}{h} + 1) - \ln \frac{\eta}{4}}{l}} \tag{2}$$

We require R^{test} to be small. Thus we want to minimise $VCdim(\mathcal{H})$.

In figure 1 can be proven for a certain arrangement of data points within a unit sphere :-

$$VCdim(\mathcal{H}) \leq \min\{R^2 A^2, N\} + 1$$

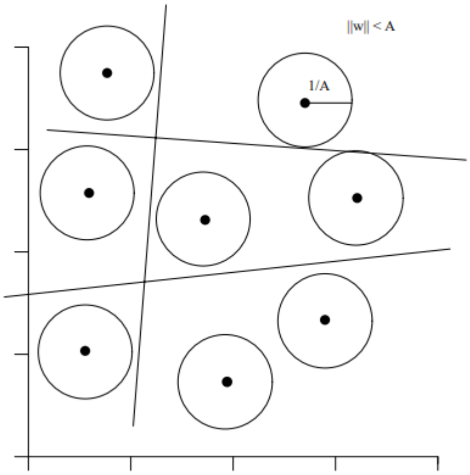

Fig. 1. Depiction of SVM classifier

In order to minimize A, we have to minimize $||\vec{w}||$. To avoid the problem of $\vec{w}$ and b becoming zero, we slightly change the constrain to $y_i(\vec{w}^\top \vec{x_i} + b) \geq 1$. This finally gives the SVM optimisation problem as:

$$\min_{\vec{w}, b} \frac{||\vec{w}||^2}{2}$$
$$\text{s.t. } y_i(\vec{w}^\top \vec{x_i} + b) \geq 1 \; \forall i = 1....n$$

The above is a convex optimisation problem, specifically, a quadratic program. In order to overcome the hurdle of linearly inseparable vectors, we allow some vectors to be misclassified by allowing some error ξ_i for each vector $\vec{x}_i$. The new optimisation problem is called the soft-margin SVM and can be formulated as:

$$\min_{\vec{w}, b, \vec{\xi}} \frac{||\vec{w}||^2}{2} + C(\sum_{i=1}^{l} \xi_i)^k$$
$$\text{s.t. } y_i(\vec{w} \cdot \vec{x} + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

This is again a convex optimisation problem. Since the problems are convex, we can also obtain the optimal hyperplane by solving the dual of the problem. The dual of the hard margin SVM is given by:

$$\max_{\vec{\lambda},\delta} \vec{\lambda}\cdot\vec{1} - \frac{1}{2}\vec{\lambda}\cdot D\vec{\lambda} \tag{3}$$

$$\text{s.t. } \vec{\lambda}\cdot\vec{y} = 0 \tag{4}$$

$$\vec{\lambda} \geq 0 \tag{5}$$

The dual of the soft-margin SVM is given by

$$\max_{\vec{\lambda},\delta} \vec{\lambda}\cdot\vec{1} - \frac{1}{2}\vec{\lambda}\cdot D\vec{\lambda} - \frac{\delta^{\frac{k}{k-1}}}{(kC)^{\frac{1}{k-1}}}\left(1-\frac{1}{k}\right) \tag{6}$$

$$\text{s.t. } \vec{\lambda}\cdot\vec{y} = 0 \tag{7}$$

$$\vec{\lambda} \leq \delta \tag{8}$$

$$\vec{\lambda} \geq 0 \tag{9}$$

Here, D is a nxn matrix with elements $D_{ij} = y_i y_j \vec{x}_i \cdot \vec{x}_j$. Optimum $\vec{w}$ and b are obtained by using the equations:

$$\vec{w}^* = \sum_{i=1}^{n} \lambda_i^* y_i \vec{x}_i$$

$$b^* = y_i - \vec{w}^\top \vec{x}\ if\ \lambda_i \neq 0$$

Another approach to overcome the hurdle of linearly inseparable vectors is by using the Kernel SVM. For this, we map the vectors to a higher dimension using the transformation $\vec{x} \to \phi(\vec{x})$. Then we apply the soft-margin SVM. In the dual form, the optimisation problem turns out exactly same as the soft-margin SVM with the exception of $D_{ij} = y_i y_j \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$. Optimum $\vec{w}$ and b are obtained by using the equations:

$$\vec{w}^* = \sum_{i=1}^{n} \lambda_i^* y_i \phi(\vec{x}_i)$$

$$b^* = y_i - \vec{w}^\top \vec{x}\ if\ \lambda_i \neq 0$$

Some commonly used kernels are

Kernel Function, $K(\vec{x},\vec{y})$	Type of Classifier
$\exp -\|\|\vec{x}-\vec{y}\|\|^2$	Gaussian RBF
$(1+\vec{x}\cdot\vec{y})^d$	Polynomial of degree d
$\tanh(\vec{x}\cdot\vec{y}-\theta)$	Multi-Layer Perceptron

II. Summary of our work

A. SVM Theory

For our project, we closely followed the theory from [1]. We first coded the hard margin SVM in the primal and dual forms. We used CVXPY to solve the obtained quadratic programs. Then we coded the soft-margin SVM in the primal and dual forms, again solving the quadratic program using CVXPY. Then we coded the kernel SVM using second order polynomial kernel and radial basis function kernel.

B. Spam Classification

For spam classification, we used pre-processed data of spam emails. The data was pre-processed by

- Lemmatizing the emails
- Creating one-hot vectors representing the frequency with which certain words appeared in the emails

The one-hot vectors become our vecctors $\vec{x}_i$. If email i was spam, we mapped the vector $\vec{x}_i$ to class 1 otherwise -1. We then applied the soft-margin SVM to this classification problem.

III. Problems faced

We faced the following problems in solving SVM and its variants in its quadratic form using CVXPY:

- It was observed that positive semi-definite matrices with any eigenvalue close to zero can become indefinite because the zero eigen value might become slightly negative due to floating point errors. To alleviate this problem, we pertubed the said matrix by adding ($\epsilon * I_{nXn}$, ϵ is a small value($1e-5$)). This will increase the eigen values by ϵ as easily seen by spectral decomposition of a positive semi-definite matrix.
- CVXPY fails for large, sparse quadratic program 689: The matrix used in the objective of the kernel SVM problem is a large matrix so due to limitations to CVXPY in handling such matrices, we had to use the psd_wrap utility provided in cvxpy.atoms.affine.wraps.
- In gaussian kernel, feature vector x is mapeed to $\phi(x)$ which maps it to infinite dimensions. In order to calculate the weight vector, we need to approximate the value of $\phi(x)$ to some dimensions. We achieved this using RBFSampler utility from sklearn.kernel_approximation. This uses Nystroem sampling from [2].

IV. Contributions

- Kartik Srinivas: Slides for empirical risk minimisation, VC dimensions in SVM, Duality.
- Tanmay Garg: Shattering in VC dimension, Spam Classification
- Aayush Patel: Slides for explaining the formulation of the convex optimisation problem of linearly separable case of SVM in primal and dual form.
- Dishank Jain: Slides for soft margin SVM, proof-reading, report.
- Gautham Bellamkonda: Slides for kernel SVM.
- Code contributions: Everyone contributed equally to the code for the three cases of SVM in primal and dual form including writing the spam classifier.

References

[1] Edgar E. Osuna, Robert Freund and Federico Girosi, Support Vector Machines: Training and applications.

[2] Ali Rahimi and Ben Recht, Random Features for Large-Scale Kernel Machine

[3] https://courses.engr.illinois.edu/ece544na/fa2014/vapnik71.pdf

[4] https://courses.cs.vt.edu/cs5824/Fall15/pdfs/

[5] https://www.cs.huji.ac.il/w~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf
[6] https://svivek.com/teaching/lectures/slides/colt/vc-dimensions.pdf
[7] https://www.youtube.com/watch?v=8yWG7fhCpTw