Available Online at www.ijecse.org ISSN- 2277-1956

Data classification using Support vector Machine (SVM), a simplified approach

S Amarappa [1] , Dr. S V Sathyanarayana [2]

[1] S Amarappa, Associate Professor
Department of Telecommunication Engg.
Jawaharlal Nehru National College of Engineering, Shimoga - 577 204
[2] Dr. S V Sathyanarayana, Professor
Department of Electronics and Communication Engg.
Jawaharlal Nehru National College of Engineering, Shimoga - 577 204

Abstract-In all our day to day activities we will be classifying things based on situations and on our needs. Human beings do classification of any kind by their natural perception. Classifying data is a common task in machine learning which requires artificial intelligence. Support vector Machine (SVM) is a new technique suitable for binary classification tasks. SVMs are a set of supervised learning methods used for classification, regression and outliers detection. The SVM classifiers work for both linear and nonlinear class of data through Kernel tricks. A Support Vector Machine is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new samples. In this paper, use of SVM for data classification is presented in a simplified way. Discussions are justified with illustrative practical examples. An effective algorithm is developed for data classification on python platform using sklearn tool kit. The results are exhibited both symbolically and graphically. This paper is expected to be an insight for desired readers and researchers in implementing their ideas of item classification using SVM.

Keywords: Support vector machines, classification, support vectors, maximum margin, hyperplane, positive gutter, negative gutter.

1 INTRODUCTION

It is known that internet is a web of big data where data belongs to different classes. Systematic storage of the data is very much essential to selectively access the required class of information which is in the form of text files, image files, audio files, video files etc. Each information type in turn belongs to different categories, for example text files may be related to sports, movies, medical imaging, genes, politics, history, geography etc. Video file may be related to education, religious, music etc. Building Machine learning models, to know, to which particular class the data belongs is interesting and challenging.

Classification is the task of choosing the correct class label for a given data input. In basic classification tasks, each input is considered in isolation from all other inputs, and the set of labels is defined in advance. Some examples of classification tasks are like (i) deciding whether an email is spam or not. (ii) Deciding the topic of a news article, is it, from a fixed list of topic areas such as "sports," "technology," and "politics."? The basic classification task has a number of interesting variants. For example, in multi-class classification, each instance may be assigned multiple labels, in open-class classification, the set of labels is not defined in advance, and in sequence classification, a list of inputs is jointly classified. A classifier is called supervised if it is built based on training corpora containing the correct label for each input. The framework used by supervised classification is shown in Figure1.

Figure1 explains the Supervised Classification: (a) during training, a feature extractor is used to convert each input value to a feature set. These feature sets, capture the basic information about each input which later is used to classify that input. Pairs of feature sets and labels are fed into the machine learning algorithm to generate a model. (b) During prediction, the same feature extractor is used to convert unseen inputs to feature sets. These feature sets are then fed into the model, which generates predicted labels.

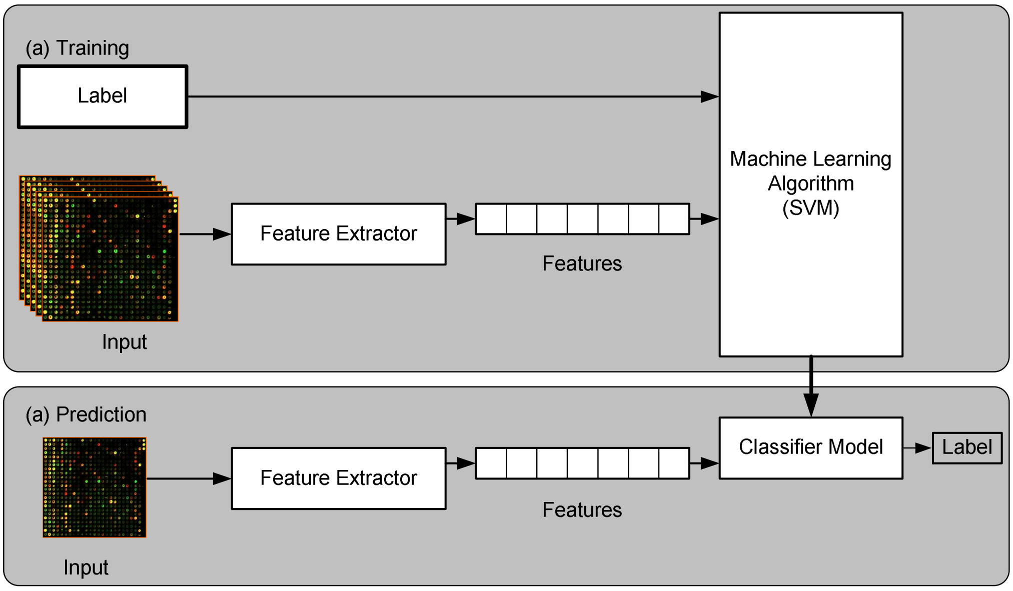

Figure1. The framework used by supervised classification

There are many algorithms which are used for classification. The most widely used classification algorithms are: (1) Rule based classification algorithms and (2) Machine Learning Classification algorithms. The Machine learning classification algorithms are of different types such as: (a) Support vector Machines (b) Artificial Neural Networks (multi-layer perceptron) (c) K- nearest neighbors (d) Gaussian mixture models namely: (i) Naive bayes classifier (ii) Decision trees (iii) RBF classifiers (iv) Hidden Markov Models.

The main challenges in classifying data include: (1) the data to be classified is often of high dimension. (2) It is hard to put up simple rules. (3) Need automated ways to deal with the data. (4) Use of computers in data processing, statistical analysis, and trying to learn patterns from the data (machine learning).

2 SUPPORT VECTOR MACHINES (SVMs)

Support vector machines (SVMs) are a set of new supervised learning methods used for binary classification, regression and outlier's detection. Among all classification algorithms SVM is strong because of its simple structure and it requires less number of features. SVM is a structural risk minimization classifier algorithm derived from statistical learning theory by Vladimir Vapnik and his colleagues in 1992. Support Vector Machines were first introduced to solve the pattern classification and regression problems.

Given some data points, each belonging to one of two classes and the goal is to decide to which class a *new* data point belong. In support vector machines, a data point is viewed as an *n*-dimensional vector, in *n*-dimensional space $\boldsymbol{R}^n$ and we want to know whether we can separate such points with an $(n - 1)$ dimensional hyper plane (Canonical plane). This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes, since in general the larger the margin the lower the generalization error of the classifier. The hyper plane is found by using the support vectors and margins. To calculate the margin, two parallel supporting hyper planes are constructed, one on each side of the Canonical plane, which is "pushed up against" the two data sets. So we choose the hyperplane such that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the **maximum-margin hyperplane** and the linear classifier it defines is known as a **maximum margin classifier;** or equivalently, the **perceptron of optimal stability**.

An SVM training algorithm builds a model of data points in space so that the data points of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using the kernel trick, which implicitly maps their inputs into high-dimensional feature spaces. More formally, an SVM constructs a hyperplane or set of hyperplanes in a high or infinite-dimensional space, which can be used for classification, regression, or other tasks.

A. MULTICLASS AND MULTILABEL SVM

Multiclass classification means a classification task with more than two classes; e.g., classify a set of images of fruits which may be mangoes, oranges, apples, or pears. Multiclass classification makes the assumption that each sample is assigned to one and only one label that is, a fruit can be either a mango or an apple but not both at the same time. Multiclass SVM aims to assign labels to instances, where the labels are drawn from a finite set of several elements. SVMs classification and decision function depends on some subset of the training data, called the support vectors.

The dominant approach for multiclass classification is to reduce the single multiclass problem into multiple binary classification problems. Common approaches of reducing a multiclass problem into multiple binary classifiers include: (i) **one-versus-the-rest** also known as **one-versus-all** strategy aims at fitting one classifier per class. If there are **n**-classes of data points then for each classifier, the class is fitted against all the other **n-1** classes and hence it requires **n** classifier models to be trained. If there are only two classes, only one model is trained. Since each class is represented by one and one classifier only, it is possible to gain knowledge about the class by inspecting its corresponding classifier. This is the most commonly used strategy and is a fair default choice. (ii) **one-versus-one** approach (Knerr et al., 1990 constructs one classifier per pair of classes. At prediction time, the class which received the most votes is selected. If **n** is the number of data classes, then **n * (n - 1) / 2** classifiers are to be constructed and each one trains data from two classes. Since it requires to fit **n * (n - 1) / 2** classifiers, this method is usually slower than one-vs-the-rest.

There is also an alternative multi-class strategy, called as multi-class SVM formulated by Crammer and Singer which casts the multiclass classification problem into a single optimization problem, rather than decomposing it into multiple binary classification problems. This method is consistent than one-vs-rest classification. In practice, on-vs-rest classification is usually preferred, since the results are mostly similar, but the runtime is significantly less.

Multilabel classification assigns to each sample a set of target labels. This can be thought as predicting properties of a data-point that are not mutually exclusive, such as topics that are relevant for a document. A text might be about any of religion, politics, finance or education at the same time or none of these. **Multi output-multiclass classification** and **multi-task classification** means that estimators have to handle jointly several classification tasks. This is a generalization of the multi-label classification task, where the set of classification problem is restricted to binary classification, and of the multi-class classification task.

B. ADVANTAGES, DISADVANTAGES AND APPLICATIONS OF SVM

Advantages of SVM are SVM is effective in high dimensional spaces. It is effective in cases where number of dimensions is greater than the number of samples. It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient. Different Kernel function can be specified for the decision function i.e SVM is versatile. Versatile: different *Kernel functions* can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

Disadvantages of SVM are if the number of features is much greater than the number of samples, the method is likely to give poor performances. SVMs do not directly provide probability estimates.

Applications: SVMs find application in various fields like Handwritten digit & character recognition, Object detection & recognition, Speaker identification, Benchmarking time, Series prediction tests, Text classification, Biometrics, Content-based image retrieval, Image classification.

3 PERFORMANCE EVALUATION TECHNIQUES

The system's performance is measured in terms of Precision (P), Recall (R) and F-measure (F). Precision can be seen as a measure of exactness or quality, whereas recall is a measure of completeness or *quantity*. In simple terms, high **precision** means that an algorithm returned substantially more relevant results than irrelevant, while high **recall** means that an algorithm returned most of the relevant results. F-measure is a measure that combines precision and recall and is the harmonic mean of precision and recall.

P = Number of correct answers - produced / *Total number of answers - produced*
R = Number of correct answer - produced / *Total number of possible- correct answers*
F -Measure = 2*PR* / (*P* + *R*). The traditional F-measure or balanced F-score

4 PRINCIPLES OF SVM

A. HYPERPLANE

In geometry, as a plane has one less dimension than space; a **hyperplane** is a subspace of one dimension less than its ambient space. A hyperplane of an **n**-dimensional space **V** is a flat subset with dimension **n – 1** in **V**. By its nature, it separates the space into two half spaces. As an example, a point is a hyper plane in **1**-dimensional space, a line is a hyperplane in **2**-dimensional space, and a plane is a hyperplane in **3**-dimensional space. A line in **3**-dimensional space is not a hyperplane, and does not separate the space into two parts.

SVM is a Machine Learning technique of classificaton and is a two-class classifier based on the use of Linear Discriminant Function **g** $(\boldsymbol{X})$ **=** $\mathbf{w}^{\mathbf{T}}\boldsymbol{X}$ **+ b**, which represents a hyperplane in the feature space. A discriminant function represents a surface which separates the patterns so that the patterns from the two classes lie on the opposite sides of the surface. The challenge is to classify given data points using a linear discriminant function in order to minimize the error rate. There are infinite numbers of answers possible, but which is optimal and best?

The linear discriminant function with maximum margin is the optimal and best solution to the above question and is obtained by SVM by determining a separating hyperplane which is optimal according to a criterion as follows: Suppose Class labels are denoted by **+1** and **-1** and **L** is a set of labeled training patterns then, $\boldsymbol{X} = \{(x_i, y_i), 1 \leq i \leq L\}$, $\boldsymbol{X_i}$ € $\mathbf{R}^{\mathbf{n}}$, $\boldsymbol{Y_i}$ € {-1, +1} and each $\boldsymbol{X_i}$ is an **n**-dimensional real vector. The SVM determine the optimal linear discriminant function with help of support vectors and is given by $\mathbf{w}^{\mathbf{T}}\boldsymbol{X}_{\mathbf{i}}$ **+ b = 0** where $\mathbf{Y_i}$ = **0** for all i. This is the equation of hyperplane. To find the maximum margin between two classes, two support planes are determined. Positive class support plane (positive gutter) is denoted by $\mathbf{w}^{\mathbf{T}}\boldsymbol{X_i}$ **+ b=+1** which lies in positive class($\boldsymbol{Y}_{\mathbf{i}}$= **+1)**. Negative class support plane (negative gutter) is denoted by $\mathbf{w}^{\mathbf{T}}\boldsymbol{X_i}$ **+b= -1** which lies in negative class($\boldsymbol{Y}_{\mathbf{i}}$= **-1)**. Here **w €** $\mathbf{R}^{\mathbf{n}}$ represents the normal to the hyperplane and **b € R** is the offset to y-axis. To maximize the margin it should **minimize** $\frac{1}{2}\|\mathbf{w}\|^2$ such that, for $\boldsymbol{Y}_{\mathbf{i}}$**=+1,** $\mathbf{w}^{\mathbf{T}}\boldsymbol{X_i}$**+b** $\geq$ **1 and for** $\boldsymbol{Y}_{\mathbf{i}}$**= -1,** $\mathbf{w}^{\mathbf{T}}\boldsymbol{X_i}$ **+ b** $\leq$ **-1.** The geometrical representation of the linear discriminant function (hyperplane) and the support planes is shown in Figure2.

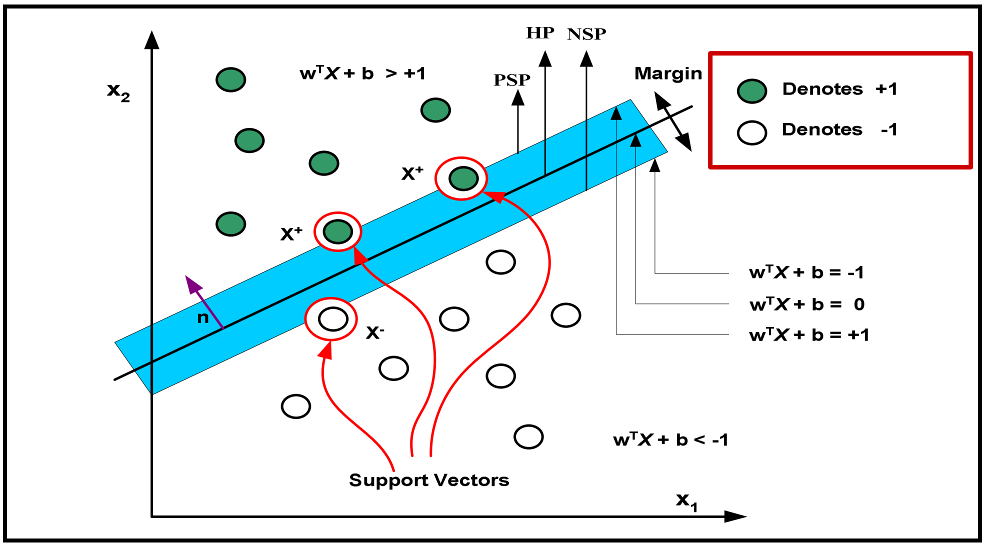

Figure2. Graphical representation of SVM principle

The SVM parameters associated with the graph are:

- **n:** is a unit-length normal vector of the hyperplane given by **n** = $\frac{\mathbf{w}}{\|\mathbf{w}\|}$, ($\|\mathbf{n}\|$ = 1)
- **w:** is the weight vector normal to the hyperplane, which is determined from alphas and support vectors.
- $\boldsymbol{X^+}$ ***and*** $\boldsymbol{X^-}$**:** Positive class support vector (SV) and negative class support vectors respectively. The nearest points of the two classes which fall on the support panes are called as support vectors.
- $X = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix}$
- **PSP, HP, NSP:** Positive support plane, hyperplane and negative support panes respectively
- Margin is defined as the width that the boundary (positive and negative gutters) could be increased (pushed apart) by before hitting a data point.

  The margin width is $\mathbf{M} = (\boldsymbol{X^+} - \boldsymbol{X^-}).\mathbf{n} = (\boldsymbol{X^+} - \boldsymbol{X^-}).\frac{\mathbf{w}}{\|\mathbf{w}\|} = \frac{\mathbf{2}}{\|\mathbf{w}\|}$
- $\mathbf{w^T}\boldsymbol{X} + \mathbf{b} = \mathbf{0}$**:** for all points on hyperplane
- $\mathbf{w^T}\boldsymbol{X^+} + \mathbf{b} = \mathbf{1}$**:** for all X^+ lying on positive gutter line (Y_i= +1)
- $\mathbf{w^T}\boldsymbol{X^-} + \mathbf{b} = -\mathbf{1}$**:** for all X^- lying lies on negative gutter line (Y_i= -1)
- $\mathbf{w^T}\boldsymbol{X} + \mathbf{b} > 1$**:** for all points in positive class (Y_i= +1)

- $\mathbf{w}^{\mathbf{T}}\boldsymbol{X}+\mathbf{b}<-\mathbf{1}$**:** for all points in negative class (Y_i= -1)
- $\mathbf{w}^{\mathbf{T}}\boldsymbol{X}+\mathbf{b}\geq\mathbf{1}$**:** for all points on positive gutter line or above (Y_i= +1)
- $\mathbf{w}^{\mathbf{T}}\boldsymbol{X}+\mathbf{b}\leq-\mathbf{1}$ **:** for all points on negative gutter line or below (Y_i= -1)

B. *GEOMETRY OF SVM*

In geometry, the equation of a straight line is given by $\mathbf{y}=\mathbf{mx}+\mathbf{b}$, which can also be written as:

$$(-\mathbf{m})\mathbf{x}+\mathbf{y}+(-)\mathbf{b}=\mathbf{0}$$

$$\mathbf{w_1x}\ +\mathbf{w_2y}\ +\mathbf{b}=\mathbf{0},\ \textbf{where}\ \mathbf{w_1}=-\mathbf{m},\ \ \mathbf{w_2}=\mathbf{1},\ \ \mathbf{b}=(-)\mathbf{b}$$

$$\begin{bmatrix}\mathbf{w_1}\\ \mathbf{w_2}\end{bmatrix}^{\mathbf{T}}\ \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}+\mathbf{b}=\mathbf{0}$$

$$[\mathbf{w_1}\quad \mathbf{w_2}]\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}+\mathbf{b}=\mathbf{0}$$

$$\mathbf{w}^{\mathbf{T}}\boldsymbol{X}\ +\mathbf{b}=\mathbf{0},\ \textbf{where}\ \boldsymbol{X}=\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}=\begin{bmatrix}\mathbf{x_1}\\ \mathbf{x_2}\end{bmatrix},\ \mathbf{w}=\begin{bmatrix}\mathbf{w_1}\\ \mathbf{w_2}\end{bmatrix},\ \mathbf{w}^{\mathbf{T}}=[\mathbf{w_1}\quad \mathbf{w_2}]$$

Here $\mathrm{w_1}$ and $\mathrm{w_2}$ are weights of x and y and b is the intercept to y-axis. Here the weights $\mathrm{w_1}$ and $\mathrm{w_2}$ may be positive or negative. If weights are +1 or -1 then slope is 1, for other weights line is same with different slopes. Plot of the line $\mathbf{w_1x}\ +\mathbf{w_2y}\ +\mathbf{b}=\mathbf{0}$ is as shown in Figure3.

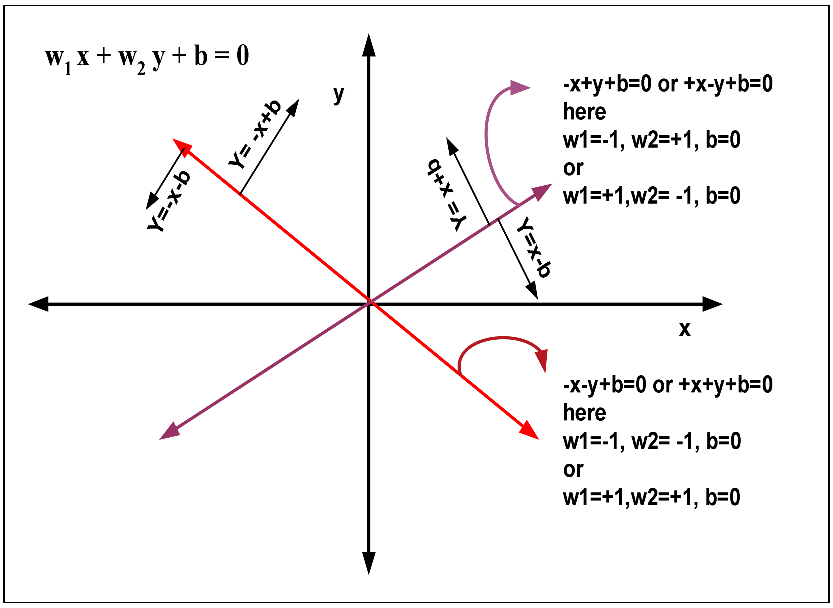

Figure3. Plot of the line $\mathbf{w_1x}\ +\mathbf{w_2y}\ +\mathbf{b}=\mathbf{0}$ for different values of $\mathrm{w_1}$, $\mathrm{w_2}$ and b

The line $\mathbf{w_1x}\ +\mathbf{w_2y}\ +\mathbf{b}=\mathbf{0}$ **or** $\mathbf{w}^{\mathbf{T}}\boldsymbol{X}\ +\mathbf{b}=\mathbf{0},$ is used as a hyperplane in two class classification problems. The hyperplane parameters w and b should be such that the line should pass through exactly in the middle of the space between two classes and the perpendicular distance of nearest points (support vectors) of two classes to the hyperplane must be same. In SVMs we are trying to find a decision boundary (hyperplane) that maximizes the "margin" or the "width of the road" separating the positives from the negative training data points and adding or deleting non-support vector points will not change the solution. To find this we **minimize**:$\frac{1}{2}\ \|\boldsymbol{w}\|^2$ subject to the constraints $\boldsymbol{Y_i}\ (\ \boldsymbol{w^T}\ \boldsymbol{X}+\boldsymbol{b})\geq\mathbf{1}$**.** The resulting Lagrange multiplier equation we try to optimize is:

Minimize $\boldsymbol{L_P}(\boldsymbol{w},\boldsymbol{b},\boldsymbol{\alpha_i}\ )=\frac{1}{2}\ \|\boldsymbol{w}\|^2-\sum_{i=1}^{n}\boldsymbol{\alpha_i}\,(\boldsymbol{Y_i}\,(\,\boldsymbol{w^T}\ \boldsymbol{X_i}+\ \boldsymbol{b})-\ \mathbf{1})$ such that $\boldsymbol{\alpha_i}\ \geq\mathbf{0}$

Solving the above Lagrangian optimization problem will give us **w, b,** and **alphas**. These parameters determine a **unique** maximal margin solution.

Useful Equations for solving SVM problems

A. Equations derived from optimizing the Lagrangian:

1. Partial of the Lagrangian wrt to b: From $\frac{\partial Lp}{\partial b} = 0$
$\sum_{i=1}^{n} \alpha_i Y_i = 0$ Note that Y_i **€ {-1, +1} and** $\alpha_i = 0$ for non-support vectors. Sum of all alphas (support vector weights) with their signs should add to 0.

2. Partial of the Lagrangian wrt to w: From $\frac{\partial Lp}{\partial w} = 0$	
$w = \sum_{i=1}^{n} \alpha_i \; Y_i \, X_i$	For when using a linear kernel. The summation only contains support vectors. Support vectors are training data points with $\alpha_i > 0$

B. Equations from the boundaries and constraints:

3. The Decision boundary:	
$h(x) = \sum_{i=1}^{n} \alpha_i \; Y_i \, K(\, X_i, X) + b \; \geq 0$	General form, for any kernel. To classify an unknown x, we compute the kernel function $K(x_i, x)$ against each of the support vectors x. Support vectors are training data points with $\alpha_i > 0$
$h(x) = \sum_{i=1}^{n} [(\alpha_i \; Y_i \, X_i). X] + b \; \geq 0$ $h(x) = \; (w^T . \; X + b) \geq 0$	For when using a linear kernel $K(X_i, X) = X_i . X$
4. Positive gutter:	
$h(x) = \sum_{i=1}^{n} \alpha_i \; Y_i \, K(X_i, X) + b = 1$	General form, for any kernel.
$h(x) = \sum_{i=1}^{n} [(\alpha_i \; Y_i \, X_i). X] + b = 1$ $h(x) = \; (w^T . \; X + b) = 1$	For use when the Kernel is linear.
5. Negative gutter:	
$h(x) = \sum_{i=1}^{n} \alpha_i \; Y_i \, K(\, X_i, X) + b = -1$	$h(x) = \; (w^T . \; X + b) = -1$
6. The width of the margin (or road):	
$width\ of\ the\ road \; \equiv \; m = \frac{2}{\lVert w \rVert} \quad where, \; \lVert w \rVert = \sqrt{\sum_{i}^{n} w_i^2}$	
Alternate formula for two support vector case: $width\ of\ the\ road \; \equiv \; m = \frac{w}{\lVert w \rVert} . (X^+ - X^-)$	This is useful when solving SVM problems in 1D or 2D, where the width of the road can be visually determined.

7. Common SVM kernels:	
a	Linear kernel ***K(u,v) = u.v***
b	Decomposable Kernels
c	Polynomial Kernel
d	Radial Basis Function (RBF) or Gaussian Kernel
e	Sigmoidal (tanh) Kernel: Allows for combination of linear decision boundaries
f	Linear combination of Kernels Idea: Kernel functions are closed under addition and scaling (by a

positive number).

Solving for alpha, b, and w by computing Kernels and solving Constraint equations

Example1: Let A = [0, 0] belong to class $\boldsymbol{Y}$ **= +1** and B = [4, 4] belong to class $\boldsymbol{Y}$ **= -1**

Step1. Determine the support vector points from the given data by finding the distance between each pair of data points from both classes. Since only two data points are given one in each class, they themselves are the support vectors. That is A = [0, 0] Support vector in positive class and B = [4, 4] Support vector in negative class.

Step2. Using linear kernel K (u, v) = u. v (dot product), compute all kernel values as follows:

K (A, A) = A.A = 0; K (A, B) = A. B = 0; K (B, A) = B.A = 0; K (B, B) = A.B = 32

Step3. System of equations using SVM Constraints

1. $\sum_{i=1}^{n} \boldsymbol{\alpha_i Y_i = 0}$; Algebraic sum of all alphas = 0
2. $\sum_{i=1}^{n} \boldsymbol{\alpha_i\ Y_i\ K(X_i, X) + b = +1}$: **+ve gutter**
3. $\sum_{i=1}^{n} \boldsymbol{\alpha_i\ Y_i\ K(X_i, X) + b = -1}$: **−ve gutter**

For support vectors A and B the above three equations reduce to:

1. $\mathbf{y_A\ \alpha_A + y_B\ \alpha_B\ = 0}$
2. $\mathbf{y_A * K\ (A,\ A) * \alpha_A +\ y_B * K(B, A) * \alpha_B\ + b = +1}$
3. $\mathbf{y_A * K\ (A, B) * \alpha_A + y_B * K(B, B) * \alpha_B\ + b = -1}$

The numerical calculations are given in Table1.

Table1: Numerical calculations of step3 equations

	Kernel value * class sign	Alpha of A	Kernel value * class sign	Alpha of B	co-eff of b	inter cept	Class label	Equations
1.	+1	α_A +	-1	α_B +	0	b =	0	(+1) α_A + (-1) α_B + (0)b = 0
2.	$Y_A * K(A, A) = +0$	α_A +	$Y_B * K(B, A) = -\ 0$	α_B +	1	b =	+1	(+0) α_A + (-0) α_B + (1)b = 1
3.	$Y_A * K(A, B) = +0$	α_A +	$Y_B * K(B, B) = -\ 32$	α_B +	1	b =	-1	(+0) α_A + (-32) α_B + (1)b = -1

Step4: Solving for $\boldsymbol{\alpha}$**'s, b** and **w** using linear algebra and augmented matrix reduction we get

R1	1	-1	0	0	From R2 we get	b = 1
R2	0	0	1	1	From R3 we get	-32 α_B + b = -1 or $\alpha_B = \frac{1}{16}$
R3	0	-32	1	- 1	From R1 we get	$\alpha_A = \alpha_B = \frac{1}{16}$

Step5: To find **w**.

We have the formula for w as: $\boldsymbol{w = \sum_{i=1}^{n} \alpha_i\ Y_i\ X_i}$

$$\boldsymbol{w = y_A * \alpha_A * A + y_B * \alpha_B * B}$$

$$\boldsymbol{w = (+1) * \left(\frac{1}{16}\right) * \begin{bmatrix}0\\0\end{bmatrix} + (-1)\left(\frac{1}{16}\right) * \begin{bmatrix}4\\4\end{bmatrix}} \quad \text{that is} \quad \boldsymbol{w = \begin{bmatrix}-0.25\\-0.25\end{bmatrix}}$$

Step6: Now the hyperplane equation $\mathbf{w_1x + w_2y + b = 0}$ becomes

$\mathbf{-0.25x - 0.25y + 1 = 0}$: Hyper plane

$\mathbf{-0.25x - 0.25y + 1 = 1}$: Positive support plane

$\mathbf{-0.25x - 0.25y + 1 = -1}$: Negative support plane

Note:

- w_1, w_2 and b are the trained parameters, and support vectors are [0, 0] and [4, 4]
- If we interchange the class of points A and B we get the same equation

The experimental plot of these planes is given in graph of Figure4 below which gives maximum margin.

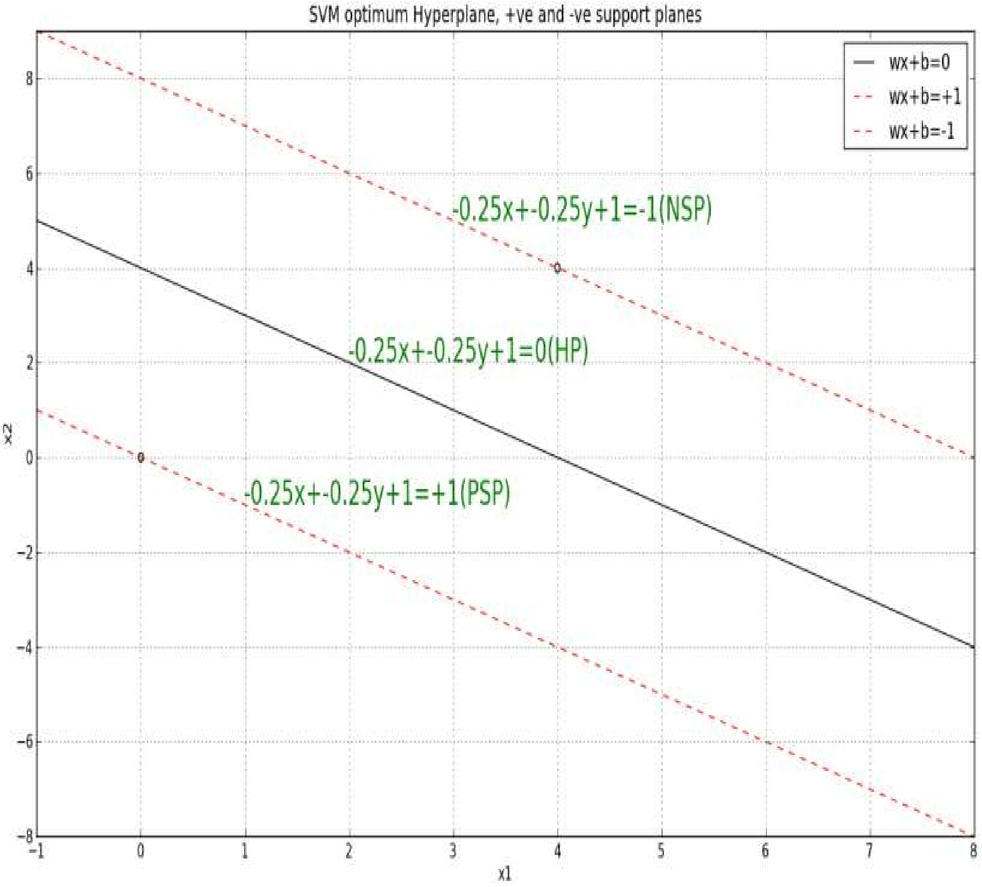

Figure4: Plot of Hyper plane, Positive and Negative support planes for Example1

Step7: Max Width (margin) = $\frac{2}{\|w\|}$ = $\frac{2}{\sqrt{\left(\frac{-1}{4}\right)^2+\left(\frac{+1}{4}\right)^2}}$ = $4\sqrt{2}$

Maximum margin minimizes $\frac{1}{2}\ \|w\|^2$ that is $\frac{1}{2}\ \|\mathbf{w}\|^2$ $= \frac{1}{2}\sqrt{\left(\frac{-1}{4}\right)^2+\left(\frac{+1}{4}\right)^2}$ $=$ $\frac{1}{16}$

Step8: Now take any test point and find to which class it belongs: say [5, 6]
Ans: Substitute this point in hyperplane equation we get $-0.25(5) - 0.25(6) + 1 = -1.75$
so the point belongs to negative class.
Now take other test point and find to which class it belongs: say [1, -4]
Ans: Substitute this point in hyperplane equation we get $-0.25(1) - 0.25(-4) + 1 = +1.75$
so the point belongs to positive class.
Example2: Let A = [2, 0] belong to class ***Y* = +1** and
Let B = [0, 0], C = [1, 1] belong to class ***Y* = -1**

Step1: Find the distance between two vectors by the formula $\sqrt{(\mathbf{x}-\mathbf{x_0})^2+(\mathbf{y}-\mathbf{y_0})^2}$
Distance between points A & B = $\sqrt{(2-0)^2+(0-0)^2} = \sqrt{4}$
Distance between points A & C = $\sqrt{(2-1)^2+(0-1)^2} = \sqrt{2}$
Distance between A and C is least and therefore A, C are the support vectors.

Step2. Using linear kernel K (u, v) = u. v (dot product), compute all kernel values for support vectors as follows:

K(A, A) = 2*2+0*0 = 4	**K(A, C) = 2*1+0*1 = 2**
K(C, A) = 1*2+1*0 = 2	**K(C, C) = 1*1+1*1 = 2**

Step3. System of equations using SVM Constraints

1. $\sum_{i=1}^{n} \alpha_i Y_i = 0$; Algebraic sum of all alphas = 0
2. $\sum_{i=1}^{n} \alpha_i \ Y_i \ K(X_i, X) + b = +1$: **+ve gutter**
3. $\sum_{i=1}^{n} \alpha_i \ Y_i \ K(X_i, X) + b = -1$: **−ve gutter**

For support vector A and C the above three equations reduce to:

1. $y_A \ \alpha_A + y_B \ \alpha_C = 0$
2. $y_A * K(A, A) * \alpha_A + y_C * K(C, A) * \alpha_C + b = 1$
3. $y_A * K(A, C) * \alpha_A + y_C * K(C, C) * \alpha_C + b = -1$

The numerical calculations are given in Table2.

Table2: Numerical calculations of step3 equations

	Kernel value * class sign	Alpha of A	Kernel value * class sign	Alpha of B	co-eff of b	inter cept	Class label	Equations
1.	+1	α_A +	-1	α_C +	0	b =	0	(+1) α_A + (-1) α_C + (0)b = 0
2.	$Y_A * K(A, A) = +4$	α_A +	$Y_C * K(C, A) = -2$	α_C +	1	b =	+1	(+4) α_A + (-2) α_C + (1)b = 1
3.	$Y_A * K(A, C) = +2$	α_A +	$Y_C * K(C, C) = -2$	α_C +	1	b =	-1	(+2) α_A + (-2) α_C + (1)b = -1

Step4: Solving for **α's, b** and **w** using linear algebra and augmented matrix reduction we get

R1	1	-1	0	0
R2	4	-2	1	1
R3	0	-2	1	-1

$\alpha_A = \alpha_C = 1, b = -1$

Step5: We have the formula for w as: $\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \ Y_i \ X_i$

$$= y_A * \alpha_A * A + y_C * \alpha_C * C$$

$$= (+1) * (1) \begin{bmatrix} 2 \\ 0 \end{bmatrix} + (-1) * (1) \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Step6: Now the hyperplane equation $\mathbf{w_1 x + w_2 y + b = 0}$ becomes

$x - y - 1 = 0$:	Or	$-x + y + 1 = 0$:	hyper plane
$x - y - 1 = 1$:	Or	$-x + y + 1 = 1$:	Positive support plane
$x - y - 1 = -1$:	Or	$-x + y + 1 = -1$:	Negative support plane

Note:

- w_1, w_2 and b are the trained parameters, and support vectors are [2, 0] and [1, 1].
- If we interchange the class of points A and C we get the same equation. The point B = [0, 0] falls on negative gutter and hence it is a support vector. To find $\mathbf{w_1}$ and $\mathbf{w_2}$ and **b** at least two SVs are required. Treating B also as SV in negative class in the beginning we could have solved for the HP equation. But by observation it is difficult to know the SVs. Hence go for minimum distance concept for finding the SVs. This is a more general way to solve SVM parameters, without the help of geometry. This method can be applied to problems where "margin" width or boundary equation can not be derived by inspection.

Example of SVMs with a Non-Linear Kernel

Example3: We are given the **positively labeled data** points at: [2, 2], [2, -2], [-2, -2], [-2, 2] in R^2. We are given the **negatively labeled data points** at: [1, 1], [1, -1], [-1, -1], [-1, 1] in R^2 and we are asked to solve for equation for the decision boundary.

Our goal, again, is to discover a separating hyperplane that accurately discriminates the two classes. Of course, it is obvious that no such hyperplane exists in the input space (that is, in the space in which the original input data live). Therefore, we must use a nonlinear SVM (that is, one whose mapping function $\emptyset$ is a nonlinear mapping from input space into some feature space).
Define:

$$\emptyset\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{cases}\begin{pmatrix}\mathbf{4} - x_2 + |x_1 - x_2|\\ \mathbf{4} - x_1 + |x_1 - x_2|\end{pmatrix} & if\ \sqrt{x_1^2 + x_2^2} \quad > 2\\ \begin{pmatrix}x_1\\x_2\end{pmatrix} & otherwise\end{cases}$$

We can see how $\emptyset$ transforms our data before the dot products are performed. Therefore, we can rewrite the data in feature space as [2, 2], [10, 6], [6, 6], [6, 10] for the positive examples and [1, 1], [1, -1], [-1, -1], [-1, 1] for the negative examples. Now we can once again easily identify the support vectors.

Let X_1 = [2, 2], X_2 = [10, 6], X_3 = [6, 6], X_4 = [6, 10] belong to class ***Y* = +1** and
Y_1 = [1, 1], Y_2 = [1, -1], Y_3 = [-1, -1], Y_4 = [-1, 1] belong to class ***Y* = -1**. In general $X_{i\,=}$ [x, y], $Y_{i\,=}$ [x_0, y_0]
Step1: Determine the support vector points from minimum distance between each point of class ***Y* = +1** and each point of class ***Y* = -1,** by the formula $\sqrt{(\mathbf{x}-\mathbf{x_0})^2 + (\mathbf{y}-\mathbf{y_0})^2}$ **.**

Distance between points X_1 & Y_1 = $\sqrt{(2-1)^2 + (2-1)^2} = \sqrt{2}$, similarly
Distance between points X_1 & Y_2 = $\sqrt{10}$, Distance between points X_1 & Y_3 = $\sqrt{18}$
Distance between points X_1 & Y_4 = $\sqrt{10}$, Distance between points X_2 & Y_1 = $\sqrt{106}$
Distance between points X_2 & Y_2 = $\sqrt{130}$, Distance between points X_2 & Y_3 = $\sqrt{170}$
Distance between points X_2 & Y_4 = $\sqrt{146}$, Distance between points X_3 & Y_1 = $\sqrt{50}$
Distance between points X_3 & Y_2 = $\sqrt{74}$, Distance between points X_3 & Y_3 = $\sqrt{98}$
Distance between points X_3 & Y_4 = $\sqrt{74}$, Distance between points X_4 & Y_1 = $\sqrt{106}$
Distance between points X_4 & Y_2 = $\sqrt{146}$, Distance between points X_4 & Y_3 = $\sqrt{170}$
Distance between points X_4 &Y_4 = $\sqrt{130}$
Distance between X_1 and Y_1 is minimum and therefore $X_{1,}$ Y_1 are the support vectors. X_1 = [2, 2] is positive class support vector and Y_1 = [1, 1] is negative class support vector. Once the support vectors are got rest of the procedure is same as in Example1.

SVM as data classifier experimental setup

Figure5 shows the block diagram of SVM classifier for data classification.

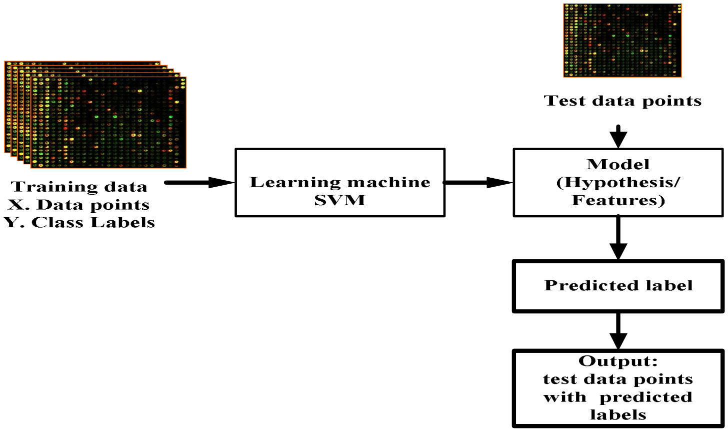

Figure5. SVM Model for Data classification

Algorithm:

Step1: Define a set of n data points in an array say
X= array([[x_{11}, x_{21}], [x_{12}, x_{22}], ... [x_{1n}, x_{2n}]])
Step2: Define class of each data point in a vector of list type say Y = [-1, -1, -1 .....1, 1, 1]
Step3: **F**it the SVM model using the statements
clf = svm.SVC(kernel='linear') and clf.fit(X, Y)
Step4: Get the separating hyperplane xx as x_1 coordinates anf yy as x_2 coordinates
w = clf.coef_[0]
a = -w[0]/w[1]
xx = np.linspace(-1, 8, 10, 1)
yy = a*xx - (clf.intercept_[0])/w[1]
Step5: Get the parallels to the separating hyperplane that pass through the support vectors
b = clf.support_vectors_[0]
yy_down = a*xx + (b[1] - a*b[0]) (positive support plane)
b = clf.support_vectors_[-1]
yy_up = a*xx + (b[1] - a*b[0]) (negative support plane)
Step6: Plot the line, the points, and the nearest vectors to the plane using appropriate python
Commands.

Conclusion

In this paper an overview of SVM is presented. This will an eyeopener for researchers in the area of data classification. SVM is basically a two class classifier. It is very encouraging as a data classifier because of its simple structure and less feature space. It can classify numerical data as well non numerical data such as text, images, patterns etc. It does multiclass classification by dividing classes into two classes one v/s all others at a time. It is to be noted that, one-versus-the-rest approach is faster than one-versus-one approach. In this paper three examples have been discussed, which will help the readers to appreciate the concept of SVM.

Bibliography

1. Scikit-learn version 0.14 documentation.
2. Piyush Rai. Hyperplane based Classification: Perceptron and (Intro to) Support Vector Machines, CS5350/6350: Machine Learning September 8, 2011.
3. Stuart Andrews, Ioannis Tsochantaridis and Thomas Hofmann. Support Vector Machines for Multiple-Instance Learning, Department of Computer Science, Brown University, Providence, RI 02912.
4. Debprakash Patnaik M.E. Introduction to Support Vector Machines (SVM), (SSA).
5. Pierre Dönnes. An Introduction to Support Vector Machine Classification Bioinformatics, Lecture 7/2/2003.
6. Wen Zhang, Taketoshi Yoshida, Xijin Tang. Text classification based on multi-word with support vector machine School of Knowledge Science, Japan Advanced Institute of Science and Technology, 1-1 Ashahidai, Tatsunokuchi, Ishikawa 923-1292, Japan, Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, PR China.
7. Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A Practical Guide to Support Vector Classification, Department of CS, National Taiwan University, Taipei 106, Taiwan, Initial version: 2003 Last updated: April 15, 2010.
8. Marina maila., SVM: A simple example, STAT 592/CSE 590 MM handout 1.1.
9. Martin Law. A Simple Introduction to Support Vector Machines, Lecture for CSE 802 Department of Computer Science and Engineering, Michigan State University.
10. Dustin Boswell. Introduction to Support Vector Machines August 6, 2002.
11. Geoffrey Hinton. Support Vector Machines, CSC 2515 2008 Lecture 10 AS/Park City Mathematics Series.
12. Richard P. Stanley. An Introduction to Hyperplane Arrangements, Volume 14, 2004.
13. VJinwei Gu. An Introduction of Support Vector Machine, 2008/10/16.
14. Radek Zíka. Support vector machines for classification.
15. Dan Ventura. SVM Example, March 12, 2009.
16. Mingyue Tan. Support Vector Machine & Its Applications, The University of British, Columbia, Nov 26, 2004.