Support Vector Machine

*An introduction of SVM and its development

Run Zeng
Department of Electrical and Computer Engineering
University Of Waterloo
Waterloo, Canada
r24zeng@uwaterloo.ca

Yewei Li
Department of Electrical and Computer Engineering
University Of Waterloo
Waterloo, Canada
yewei.li@uwaterloo.ca

***Abstract*—Because of the solid statistical learning theory and its excellent performance in small size of data sample, SVM becomes one of the most popular supervised classification algorithms. However, with the need of more and more complicated application, the standard SVM cannot satisfy our requirements any more, such as semi-supervised problem, unsupervised problem and multi-classification problems. In order to apply SVM widely in our real word, many methods based on SVM were proposed. S^3VM is used to deal with not full labeled data classification. Support Vector Clustering(SVC) can classify unlabeled dataset. One-versus-Rest(OvR) and One-versus-One(OvO) give us idea to do multi classification. DAGSVM develops OvO to reduce time complexity. Firstly, we introduce the statistical principle of SVM in detail. Then the popular extensions of SVM will be briefly explained. Finally, we give conclusion about these developments and issues still existed.**

***Index Terms*—Hard margin, soft margin, S^3VM, SVC, OvR, OvO, DAGSVM**

I. Introduction

Support Vector Machine has been a very powerful classifier among all the classification methods in the field of Machine Learning, especially before the emergence of CNN(Convolutional Neural Network). Apart from Hard-Margin SVM, Soft-Margin SVM gives SVM great generalization performance make it handle unseen data. Parameter C and Non-Linear SVM with its Kernel functions make SVM even more versatile to solve non-linear separable classification problems.

optimization, the sparf the solution, nonlinear and generalization.In our paper, we discuss the basic theroy of SVM, derived the equation of how to define the problem of SVM and use it to get the optimal decision boundary. In recent decades, the ways of collecting data are more diverse, and the amount of data is also growing. Along this change, different data throw out new challenges to traditional classifiers. In this process, based on the original SVM algorithm, it keeps evolving and adapt to new difficulties in many ways.

SVM is a very robust machine learning method. Originally, SVM has many limitations. The emergence of kernel SVM makes SVM solve non-linear problems. Another drawback for original SVM is, it can only solve supervised problems which means that all data must has its label. However, in many practical conditions, unlabeled data are always account for a big portion. It is difficult to make a classifier with strong generalization ability by using limited labeled data in supervised learning. To deal with such problem, semi-supervised SVM and unsupervised SVM came out. Semi-supervised SVM tried to rebuilt labels for the unlabel data. Unsupervised svm works without labels and try to cluster the data have similar characteristics. SVM is a bi-class classification when it was designed. However, in reality, we have to deal with many multi-class classification probelm. By two main strategies One-versus-Rest and One-versus-All, multi-class classification can also be solved.

For readers' convenience, we introduced concept of SVM mainly in section II. In section III, we talked about the newer endeavor of extensions of SVM which include using SVM to deal with semi-superviesd and unsupervised problems. And the concept using SVM to deal with multi-class classification.

II. Technical Background

A. How to find the optimal decision boundariy

There already exists so many classifiers available including kNN, Logistic Regression, Decision Trees and more. The concept for SVM is to find a decision boundary is different from other classification methods. Let's first look at some decision boundaries made by some machine learning methods.

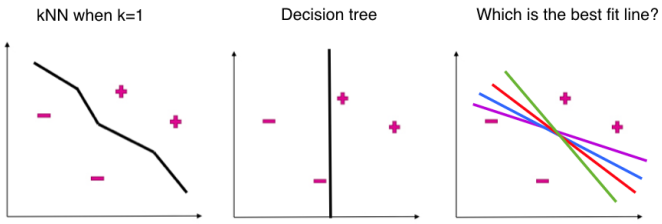

Fig. 1. Different decision boundaries for different classifiers

Suppose we have two classes: negative and positive data points. The first graph in Figure1 simulate the decision boundary generated by kNN classifier when k=1. While the decision boundary for Decision Tree is going to look like the second graph. Most of time, if the data is linearly separable that among various classifiers there exist more than one decision boundaries. Which one is the optimal decision boundary becomes an important question we have to consider when we decide to use a classifier.

We use generalization capacity as a major criterion to evaluate the performance of decision boundaries. The decision boundaries with poor generalization ability are more prone to overfit, which means the model performs well on training data, but perform poorly on test data. When it comes to new data, it might fail to split new samples correctly.

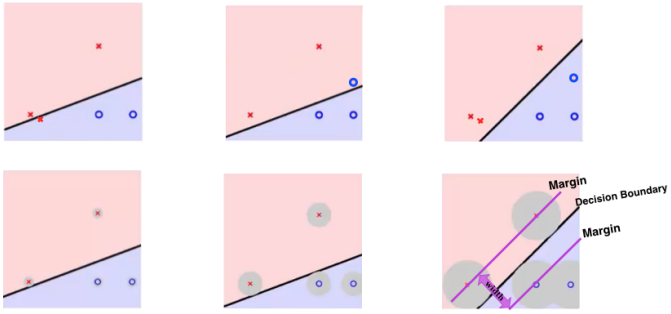

Fig. 2. An example to show the ability for different decision boundaries when it comes to unseen data

Here is an example as shown on Figure 2, from the first decision boundary we get. the sample will be labeled as blue circle. this is a miss-classification because the true label of this sample is red cross. The second decision boundary will again miss-classifies the blue circle to a red cross. However, the third decision boundary separates these two same samples perfectly. So, the 3rd model has better generalization ability.

The observation from this example is that from left to right the radius around the data points to the decision boundary is increasing. we call the distance as margin of the classifier. If we are able to find a margin separates both classes, and the width of this margin is as much wide as possible. We believe it will get the best generalization performance. This is the idea of SVM and the decision boundary is in the middle of margins.

B. Define the equation for the decision boundary in SVM

Assume we have a sample u, if we are able to get the distance between the sample and decision boundary. We can tell whether it belongs to positive or negative class. Suppose we know the distance from origin to decision boundary. W is a unit vector perpendicular to decision boundary. Dot product W times U gives us the projection of actual position of u in the direction of W. Based on the value of dot product is greater of less than c, we are able to tell the class label of sample U.

Sample is positive:

$$\vec{w} \cdot \vec{u} \geq c$$

Sample is negative:

$$\vec{w} \cdot \vec{u} < c$$

We call c a scaler and move it to the left hand side and name it as b. So the objective is to find the w and b, if we have w and b, then we have the SVM classifier.

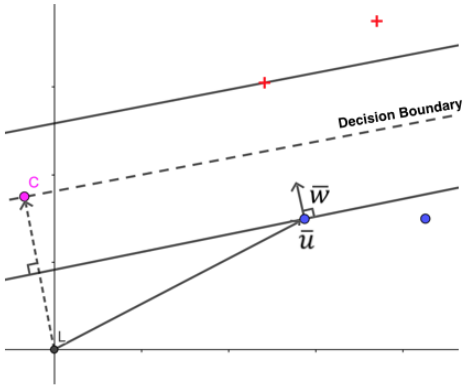

Fig. 3. A coordinate system

Every time we have the new sample, we just have to do the dot product by w plus b. If the value is great than 0 its positive, otherwise is negative class.

Sample is positive:

$$\vec{w} \cdot \vec{u} + b \geq 0$$

Sample is negative:

$$\vec{w} \cdot \vec{u} + b < 0$$

In order to facilitate our calculation. We choose to restrict more to use greater than 1 and less than minus 1 to denote two classes.

Sample is positive:

$$\vec{w} \cdot \vec{u} + b \geq 1$$

Sample is negative:

$$\vec{w} \cdot \vec{u} + b < -1$$

Also, for the purpose to combine these two conditions together. by introducing yi which has value of 1 for positive samples and -1 for negative samples.

To combine those two equations:

$$\textbf{yi s.t. yi} = +1 \textbf{ For positive class}$$

$$\textbf{yi s.t. yi} = -1 \textbf{ For negative class}$$

We multiply yi with the equations and get the following equation for both positive and negative samples:

$$\textbf{yi}(\vec{w} \cdot \vec{x} + b) \geq 1$$

$$\iff \mathbf{yi}(\vec{w} \cdot \vec{x} + b) - 1 \geq 0$$

Thus, the equation equals 0 means the samples on the margin. If result greater than 0 means the samples outside margin.

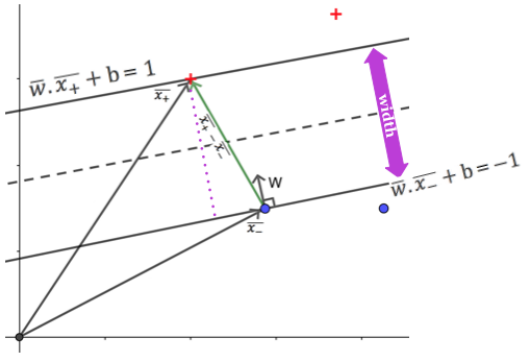

Fig. 4. Find a equation to denote the margin

For SVM we are trying to maximize the margin. The decision boundary is the middle line between margins. So we need to find an equation for the width of margin and try to maximize it. This can be expressed by the projection of the green vector(X+ - X-) to the direction of unit vector w.

$$\textbf{Width=}\frac{\vec{w}}{||\vec{w}||} \cdot (\vec{X_+} - \vec{X_-})$$

$$\iff \textbf{Width=}\frac{1-b-(1-b)}{||\vec{w}||} \cdot (\vec{X_+} - \vec{X_-}) = \frac{2}{||\vec{w}||}$$

$$\iff \textbf{Max Width=Max}\frac{2}{||\vec{w}||}$$

Transfer the maximum problem to a minimum for our convenience.

$$\textbf{Max}\frac{2}{||\vec{w}||}$$

$$\iff \textbf{Max}\frac{1}{||\vec{w}||}$$

$$\iff \textbf{Min}\frac{1}{2}||\vec{w}||^2 \ (1)$$

$$\mathbf{yi}(\vec{w} \cdot \vec{x} + b) - 1 = 0 \ (2)$$

Now we need to do minimization while under the decision boundary condition. this can be solved by Lagrange multiplier.

$$\mathbf{J(w,b) = L = \frac{1}{2} \| w \|^2 - \sum \alpha^i [y_i(\bar{w}.\bar{x_i} + b) - 1]}$$

$$\frac{\partial L}{\partial \bar{w}} = \bar{w} - \sum \alpha^i y_i x_i = 0 \longrightarrow \bar{w} = \sum \alpha^i y_i x_i$$

$$\frac{\partial L}{\partial b} = -\sum \alpha^i y_i = 0 \longrightarrow \sum \alpha^i y_i = 0$$

$$\mathbf{L = \sum \alpha^i - \frac{1}{2}(\sum \sum \alpha^i \alpha^j y_i y_j x_i \cdot x_j)}$$

By taking the derivitive regarding to w and b. we can get two new equations. substitute these two equation to make the minimize simpler. Through complex mathematical deduction. The optimization problem can be solved with numerical analytic method. Thus we can get all the values of α_i α_j.

$$\bar{w}.\bar{u} + b$$
$$\sum \alpha^i y_i \bar{x_i} \cdot \bar{u} + b \geq 0$$
$$\leq 0$$

When we have a new sample u. we can get the submission. based on the result, we are able to get the class label. If the value is greater and equal than 0, we say the class is positive. otherwise, the class is negative.

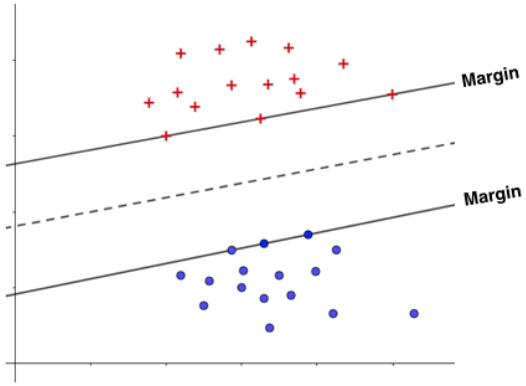

Fig. 5. Support Vectors

The points lie on the margin are called support vectors. The classifier is name after support vectors. The support vectors have α_i greater than 0 and the other points have α_i equal 0 Only the support vectors contribute to defining the decision boundary. Therefore the complete dataset could be replaced by only the support vectors while you get the same result. This also helps with reducing the computation complexity. This means that for the new sample you want to classify, you just need to figure out which samples are on the margin, use those do the submission and get the class label.

C. Limitations of Hard-Margin SVM

According by our definition so far. All data points are either on the margin or outside the margin. There are no data points fall inside the margin. We call this Hard-margin SVM. Hard-margin SVM is very sensitive to outliers. As we can see on the Figure8, one specific outlier will affect decision boundary dramatically.

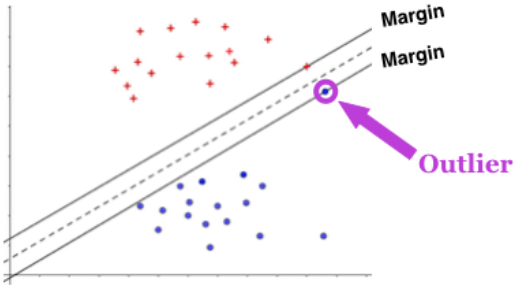

Fig. 6. Hard-Margin SVM is very sensitive to outliers

Which will again impair generalization. Another problem is that we can only deal with linearly separable dataset. If the data is not linearly separable.

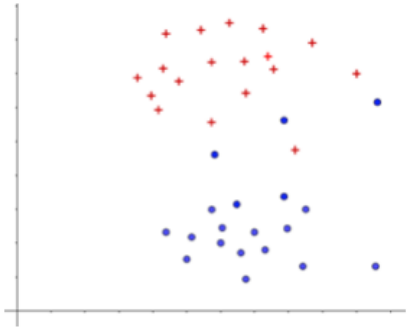

Fig. 7. An example shows the 2 classes of data is not linearly separable

There will be no solution to get a linear decision boundary to split our data.

D. Soft-Margin SVM

Soft-margin SVM came out to overcome this problem by adding slack variables ζ_i that relax the constraint. One of the equations we draw to deal with the Hard-Margin SVM problem is:

$$\mathbf{y^i}(w^T \cdot x^i + b) \geq 1$$

After adding slack variables ζ_i:

$$\mathbf{y^i}(w^T \cdot x^i + b) \geq 1 - \zeta^i \ \textit{For all i=1...N}$$

Then the previous optimization problem becomes like this.

$$\text{Minimize} \quad J(w) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{Subject to} \quad \begin{cases} y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i & \forall i \\ \xi_i \geq 0 & \forall i \end{cases}$$

Fig. 8. Equation after adding slack variable zeta

It can also be solved by lagrangian multiplier. On the other hand, in Soft-margin SVM, there is a free parameter C to control the trade-off of how much miss-classification we can tolerate.

Here's two images illustrate the effect of parameter C.

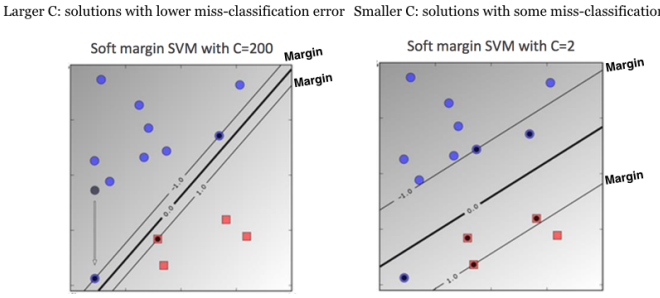

Fig. 9. Effect of different parameter C

The larger C, the smaller the margins are, the lower miss-classification error we get. It's more like Hard-margin SVM because as you can see, there is no data points falls inside the margin. The smaller value of C, the wider the margins are, the more data allowed to be fell inside the margin and with some degree of miss classification under acceptable tolerance. So that we still able to get a decision boundary and the possible widest margin which is a more generalized model. parameter C is commonly determined through cross-validation to find the most suitable C which generalize all data and provide as much higher training accuracy as possible.

Because of the C parameter, Soft-margin SVM has the ability to deal with outliers and is able to tackle some non linearly separable problems.

E. Non-Linear SVM

Feature selection and Feature extraction methods in the field of Machine Learning try to reduce the number of Dimension.

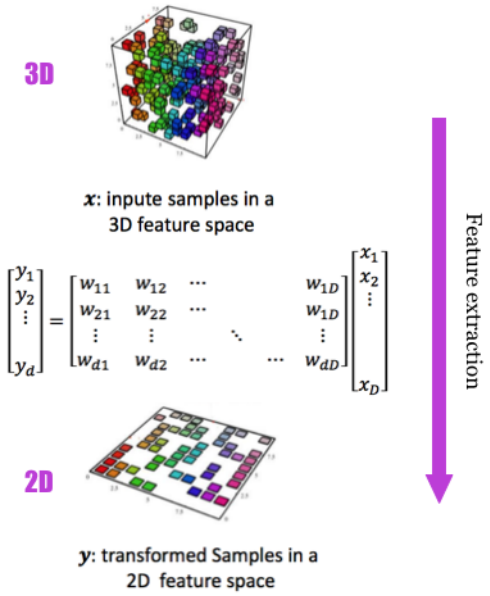

Fig. 10. Feature extraction method to reduce dimensionality

However, for the linear non-separable data, If we find a Non-linear function to project data from lower dimension to higher dimension space, there is a possibility to solve the problem by linear svm. Suppose we have a 2d data. It's impossible for us to find a linear line to discriminate the 2 classes. If we use the shown non-linear function to project the 2d samples to 3d space.

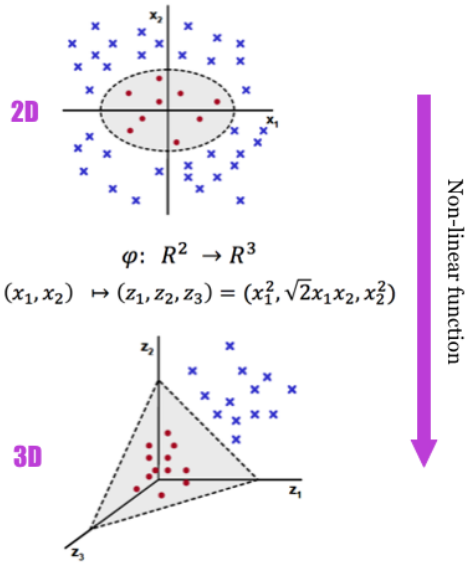

Fig. 11. Non-linear function map data to higher dimension

We can easily discriminate the mapped data with a linear decision boundary. After projection to higher dimension by the non-linear function, we are able to solve such non-linear problem with linear SVM as well.

$$h(x) = sign(w^T\varphi(x) + b) = sign\left(\sum_{i=1}^{m} \alpha_i y^{(i)} \varphi(x^{(i)})^T \varphi(x) + b\right)$$

Fig. 12. Decision function of Non-linear SVM

But the solution that mapping data to higher dimension has two issues. First issue we have is along the increasing dimension, we will need more samples to train the model. Second issue is that we are going to increase computation complexity significantly. Luckily, there exists some functions that have the same dot product result without project data in higher dimension. So we don't have these issues any more. such functions are called kernel functions. We just input two samples to the kernel function, get the same dot product result in higher dimension back.

$$K(x^{(i)}, x^{(j)}) = \varphi(x^{(i)})^T \cdot \varphi(x^{(j)})$$

$$h(x) = sign\left(\sum_{i=1}^{m} \alpha_i y^{(i)} K(x^{(i)}, x) + b\right)$$

Fig. 13. Kernel SVM is Non-Linear SVM

There are group of people in ML community working on designing new kernels for different data. And the kernel tricks are not just for SVM, it can be used for different classifier as well. But it works very well for SVM.

III. Extensions Of **SVM**

SVM is a strong classification algorithm in theory. It is easy understanding and has excellent performance in supervised problem and binary classification problem. But when we apply it to our real world, we hope to solve more complicated problems by **SVM**, such as unsupervised, semi-supervised and multi-classification problems. I will mention most popular solutions in the following paragraphs.

A. Semi-supervised learning based on SVM($\mathbf{S^3VM}$)

Most of cases, the process of labelling data for specific purpose is very slow, costly and inefficiently. For example, the full labelled sample is very small in gene expression-based prediction of cancer [1]), because micro-array experiments are time consuming, expensive and limited by data availability. But the unlabeled gene samples are abundant and easy to obtain. Therefore, we hope to take full advantages of both labeled and unlabeled database to improve the generalization performance.

The main idea is to do cluster assumption for unlabeled data for semi-supervised learning, then treat these unknown data as additional optimization variables to associate labeled data to construct classifier.

In 1999, Transductive Support Vector Machines (**TSVM**) was proposed by Joachims, this algorithm performs especially well in text classification [2]. As this method requires to assume the classification of unknown data according to classification proportion of labeled data by standard inductive learning, it still has poor ability in generalization of new data set if training data is not large enough. Then this method is improved by Chen et al [3]. in 2003 as Progressive Transductive Support Vector Machines (**PTSV**). **PTSV** gives possible label to unlabeled data in training process and retrains both labeled and unlabeled data sets. This requires labeling the unlabeled dataset frequently so that time complexity increases a lot.

As the same time as the proposal of **TSVM**, Semi-supervised Support Vector machines ($\mathbf{S^3VM}$) is proposed by Bennett and Demiriz in 1999. Its principle is also transductive inference, but it is more general in dealing with small

imbalanced training data with large unknown working set data. Similar to standard **SVM**, $\mathbf{S^3VM}$ requires to minimize misclassification error and capacity of classification function. The difference between them is that the classification function depends on both working data and training data.

Considering binary classification, the training set is defined as $\{(x_i, y_i)_{i=1}^{l}, y_i = \pm 1\}$, and u is unlabeled data which is defined as $\{x_i\}_{i=l+1}^{n}$, with $n = l + u$. The basic form of semi-supervised support vector machine can be defined as:

$$\min \Phi(\vec{w}) = \min_{\vec{w},b}\{\frac{1}{2}||\vec{w}||^2 + C_1 \sum_{i}^{l} \max(1 - y_i(\vec{w} \cdot x_i + b), 0) + C_2 \sum_{i=i+1}^{l+u} \max(1 - |\vec{w} \cdot x_i + b|, 0)\}$$

Here, C_1 and C_2 are the weight of two loss function. $i = l + 1, l + 2, \ldots, l + u$ is the unlabeled data. Based on above conception, adding two constraints to avoid that those unlabeled data would be divided into a same class, it is $\mathbf{S^3VM}$. One constraint assumes a unlabeled data belongs to class $math1$ then calculates the misclassification rate, the other constraint assumes this data belongs to class $math-1$ then calculates the misclassification rate. The $\mathbf{S^3VM}$ can be defined as: [4]

$$\min_{\vec{w},b,\eta,\xi,z} C\left[\sum_{i=1}^{l}\eta_i + \sum_{j=l+1}^{l+u}\min(\xi_j, z_j)\right] + ||\vec{w}||$$

$$\text{subject to} \quad y_i(\vec{w} \cdot x_i + b) + \eta_i \geq 1 \quad \eta_i \geq 0 \quad i = 1, \ldots, l$$

$$\vec{w} \cdot x_j - b + \xi_j \geq 1 \quad \xi_j \geq 0 \quad j = l + 1, \ldots, l + u$$

$$-(\vec{w} \cdot x_j - b) + z_j \geq 1 \quad z_j \geq 0 \quad d_j = \{0, 1\}$$

where $c \geq 0$ is a fixed misclassification penalty. If the point is in class 1 then $d_j = 0$, if it is in class -1 then $d_j = 1$. This method can solve linear problem, once encountering non-linear problem, we can transfer it to linear problem by kernel function. There are other methods to solve semi-supervised problem such as $\mathbf{LapSVM}$, $\mathbf{meanSVM}$ based on cluster kernel and so on [5].

*B. Unsupervised learning based on SVM(**SVC**)*

In classification, we know the predefined classes and want to know which class a new data belongs to. In clustering, we try to group possible related data with respect to the certain attributes based on specific criterion. Clustering may proceed in **k-means** algorithm [6], self-organizing feature maps (**SOFM**) [7] and so on. They still have limitation in types of data sets, outliers and high-dimensional data sets. However, **SVM** can overcome these drawbacks as Kernel functions can solve high dimension problem and soft margin can ignore outliers. Based on the advantages of **SVM**, **SVC** is proposed by Ben-Hur, Horn, Siegilmann, and Vapnik [8] to cluster data sets. The principle of **SVC** is, simply to say, mapping a set of data from data space to a high-dimensional feature space by Gaussian Kernel, where we search for minimal enclosing sphere. This sphere, when maps back to data space, can separate into several contours of data points. These contours are interpreted as cluster boundaries. Why is this method named Support Vector Clustering? Because only data points on cluster boundaries contribute to the result.

Next, I will describe how to find the cluster boundary through computation.

Let $x = \{x_1, x_2, \ldots, _n\} \in \mathbb{R}^n$ be the data space. Using nonlinear transformation Φ to map data points to high-dimensional feature-space, in which we are looking for the smallest enclosing sphere. $\Phi(x_j)$ describes every point in this space. This concept is described by the constraints as:

$$||\Phi(x_j) - a||^2 \leq R^2, \quad \forall j = 1, \ldots, n$$

where $||\cdot||$ denotes the Euclidean norm and a and R are the center and the radius of the sphere respectively. To deal with outliers, this constrain can be extended by adding slack variable ξ_j. Then the soft constrain can be defined as:

$$||\Phi(x_j) - a||^2 \leq R^2 + \xi_j, \quad \forall j = 1, \cdots, n, \ \xi_j \geq 0$$

To solve this problem, we introduce the Lagrangian:

$$L = R^2 - \sum_j (R^2 + \xi_j - ||\Phi(x_j) - a||^2)\beta_j - \sum_j \xi_j\mu_j + C\sum_j \xi_j$$

where $\beta_j \geq 0$ and $\mu_j \geq 0$ are Lagrange multipliers, C is a constant, $C\sum_j \xi_j$ is a penalty term. By eliminating the variable R, a and μ_j, according to Tax and Duin [9], we can rewrite Lagrangain to Wolfe dual form as follows, the kernel functions we use is Gaussian Kernel:

$$W = \sum_j K(x_i, x_j)\beta_j - \sum_{i,j} \beta_i\beta_j K(x_i, x_j)$$

$$K(x_i, x_j) = e^{-q||x_i - x_j||^2}, \quad 0 \leq \beta_j \leq C, \ = 1, \ldots, n$$

Then the radius of the sphere is:

$$R = \{R(x_i) | x_i \textit{ is a support vector}\}$$

As we know, the distance of each points image in feature space from center of sphere is:

$$R^2(X) = ||\Phi(X) - a||^2$$

The contour of enclosing the point set clustering is:

$$\{X | R(X) = R\}$$

The pairs of x_i and x_j whose images lie inside or on the sphere can be defined as matrix A_{ij}:

$$A_{ij}= \begin{cases} 1, \forall y \text{ on the line segment connecting } x_i \text{ and } x_j, R(y) \leq R \\ 0, otherwise \end{cases}$$

Therefore, every data can be assigned a cluster excluding outliers. There are two critical parameters: q, the scale parameter of the sphere, and C, the soft margin constant. Figure 14 shows that increasing q will increase the number of clusters. Figure 15 shows that C can affect the smooth of contours of clustering.

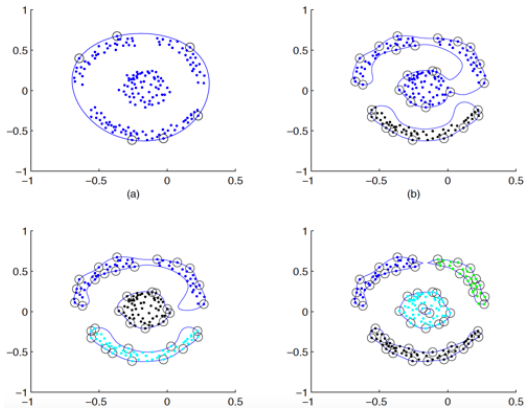

Fig. 14. [8]: Clustering 183 points by **SVC** with $C = 1$. (a) $q = 1$ (b) $q = 20$ (c) $q = 24$ (d) $q = 48$.

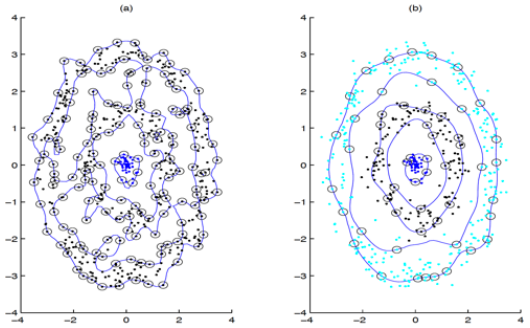

Fig. 15. [8]: (a) $C = 1, q = 3.5$, not allow outliers (b) $p = 0.3, q = 1.0$, allow outliers.

SVC is perfectly applied to Marketing Segmentation to adjust market strategies according to analyze customers. It is easy to see what kind of sale strategy or product satisfy what kind of customers in mixed data. According to the case of a drink company [10], SVC outperforms to K-means and SOFM methos.

C. Multi-class solution based on SVM

Although **SVM** outperforms to many classification algorithms, the original SVM is designed for binary classification problems. In our real life, we always want to recognize many categories at the same time, such as written digits recognition [11] and gene expression cancer diagnosis [12]. In order to take full advantage of high accuracy of **SVM** in binary classification, we develop **SVM** as Multi-class **SVM**. The main idea is to decompose multi-class classification problem to binary classification and then implement **SVM**. I will briefly introduce the principles of the widely used three methods: One-versus-Rest (**OvR**), One-versus-One (**OvO**), **DAGSVM**.

*1) **OvR**:* **OvR** was first proposed by Vladimir Vapnik [13]. The main idea is to make one class separated from rest to train data set, in other words, set one class as positive class and others as negative class. During test process, If one data is accepted by one class but rejected by rest classes, then this data belongs to the class. For example, we have A, B, C four classes. Which means we get $(A, BC), (B, AC), (C, AB)$ three classifiers after training via **SVM**. In this method, about 25% data is not assigned to any class as Figure 16 [14].

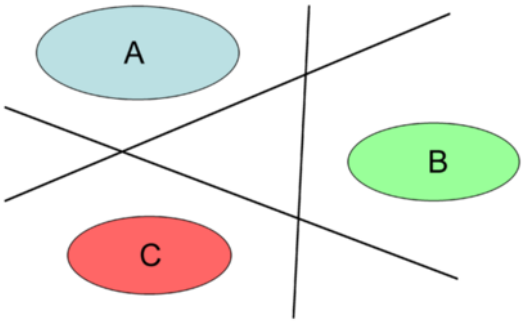

Fig. 16. [14]: binary boundaries of OvR based on A, B, C classes.

In order to solve this, the decision pattern was developed by Vaink in 1998. If we have n classes, then we have n classifiers so as n decision functions. The test data will belong to the class which achieve the highest value of decision function. The decision functions can be expressed as:

$$f(x) = \text{sgn}(\sum_{i=1}^{r} \alpha_i y_i k(\vec{x}, \vec{x_i}) + b)$$

where $\alpha_i, i = 1, \ldots, n$ are Lagrange multipliers, α_i and b can be found by maximum margin. y_i are the labels of α_i. The decision result can be expressed as:

$$\text{class of } x_i = \arg \max_{i=1,\ldots,n} ((\vec{W}_i)^T \phi(x_i) + b_i)$$

For this method, obviously, we should construct n classifiers. This development can achieve approximately 81% correct [14].

*2) **OvO**:* Another method **OvO** pairs every two classes, in other words, it sets one class as positive class and the other one as negative class. For example, three classes A, B, C can be paired as $(A, B), (B, C), (A, C)$. Let **SVM** trains each pair to get a classifier. In testing process, Max-win strategy is used. If x_i is classified in class A in classifier of (A, B), then A gets one vote. If x_i belongs to class B in classifier of (B, C), then B gets one vote. If x_i belongs to class A in classifier (A, C), the vote of class A increases from one to two. B gets one

vote, A gets two votes. Therefore, x_i is class A. The model of this method is showed as Figure 17.

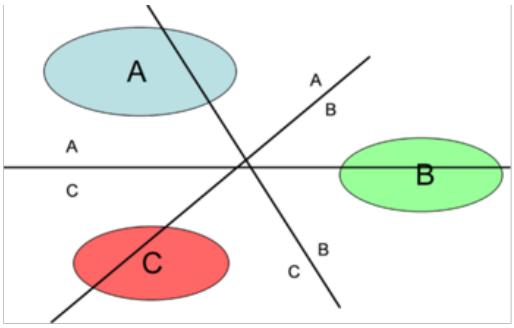

Fig. 17. [14]: binary boundaries of OvR based on A, B, C classes.

*3) **DAGSVM**:* If there are n classes, we should construct $(n(n-1))/2$ classifiers. Compared to **OvR**, for every classifier, training sample used is smaller as only choose samples from two of all classes to train, but the test process takes longer time especially there are very many classes. **DAGSVM** was proposed to overcome the disadvantage. **DAG** is a kind of graph, very similar to decision tree. Every node is a **SVM** by pairing one pair of classes. It has n leaves correspond to n classes. The training process is as the same as **OvO**, the testing process is very different, which makes testing time shorted and guarantees any data can find a class. During testing, pass every test data from root to node until reach a non-leave node to specify a class. **DAGSVM** can be described as Figure 18.

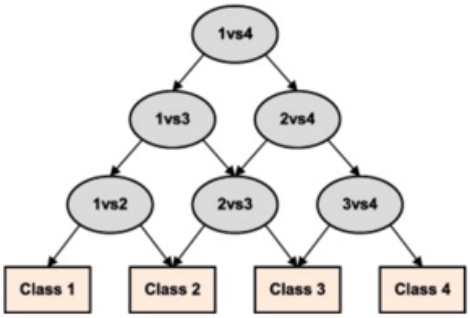

Fig. 18. [15]:DAGSVM with four classes.

For n classes, only $n-1$ nodes are needed to make decision. The number of tests is reduced. So this method is more efficient. There are still other extensions of **SVM** performing very well in solving multi-class problem, such as Binary Tree of **SVM** [16].

IV. Conclusion

It is known that **SVM** is based on minimizing the VC dimension and structural risk. The main goal is to maximize margin to separate sample in two classes. According to explanation of the mathematics principle behind **SVM**, we can see three most obvious advantages comparing to other machine learning algorithms. Firstly, because we can map data to high dimension by kernel functions, linear non-separate question can be solved. Secondly, because we only use support vectors to decide the boundary, deleting and adding other data do not affect our model. Thirdly, Soft margin **SVM** can deal with outliers to improve classification performance properly. Its limitation is also obvious, such as that it can only apply to supervised problem and binary classification problem.

Based on its both advantages and disadvantages, many extensions have been proposed to develop this algorithm. This paper gives briefly introduction of popular solutions to common issues in our real life based on **SVM**. We mainly do survey about these extensions from following two aspects.

Firstly, considering issues we encounter in practical application and the principles of solutions. For example, we can not get full labeled data sample in a short time but we want to evaluate new data. **S^{3}VM** assumes the class of unlabeled data and calculates the minimal misclassification error rate by subjective function. Then it combines both labeled and unlabeled data to train classifier. **SVC** focuses on unsupervised problem. When we do not know the clear classes of data, we still want to separate them to several classes to make personalized decision. **SVC** uses the clustering conception to cluster similar data to be one class. Multiclassification problem is still common in our life such as text classification, image recognition and so on. **OvO** and **OvR** transfer this problem to binary classification. **DAGSVM** develops **OvR** to improve efficiency and accuracy.

Secondly, considering the relations between these extensions and **SVM**. Except labeling the unlabeled data and taking all data into training process, **S^{3}VM** still uses the same strategies to maximize margin between two classes. To **SVC**, it uses kernel function to separate dataset in low data space as **SVM**. When data maps to high dimension space, they can be clustered so to be separated to a few classes. For Multiclassification **SVM**, only how to use dataset reasonably is different from **SVM**. No matter what data dealing method is used, we always transfer multi-classification to binary classification, then apply **SVM** to train corresponded classifiers respectively.

Except the main problems we discussed about in this paper, these methods can be used flexibly. For example, Combination of **S^{3}VM** and Multiclassification **SVM** can solve semi-supervised multiclassification problem. **SVC** is a good clustering tool to do data preprocessing to reduce features. Combination of **DAG** and **OvO** can improve performance of single **OvO**. In this paper, we briefly introduced the history of every method based on **SVM** and took one of the most classic methods as example to explain how they work. But we did not do more research in analyzing the generalization capacity and time complexity of every method. To different practical cases, different methods perform differently in these above two aspects. It is difficult to find one method accommodate every problem. Disadvantages and advantages of these algorithms and their scope of application need further research, which help us to find method should be used in a certain case.

REFERENCES

[1] M. Shi and B. Zhang, "Semi-supervised learning improves gene expression-based prediction of cancer recurrence," *Bioinformatics*, vol. 27, no. 21, pp. 3017–3023, 09 2011. [Online]. Available: https://doi.org/10.1093/bioinformatics/btr502

[2] T. Joachims, "Transductive inference for text classification using support vector machines," in *Icml*, vol. 99, 1999, pp. 200–209.

[3] Y.-S. Chen, G.-P. Wang, and S.-H. Dong, "A progressive transductive inference algorithm based on support vector machine," *Journal of Software*, vol. 14, no. 3, pp. 451–460, 2003.

[4] K. P. Bennett and A. Demiriz, "Semi-supervised support vector machines," in *Advances in Neural Information processing systems*, 1999, pp. 368–374.

[5] S. Ding, Z. Zhu, and X. Zhang, "An overview on semi-supervised support vector machine," *Neural Computing and Applications*, vol. 28, no. 5, pp. 969–978, 2017.

[6] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[7] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[8] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," *Journal of machine learning research*, vol. 2, no. Dec, pp. 125–137, 2001.

[9] D. M. Tax and R. P. Duin, "Support vector domain description," *Pattern recognition letters*, vol. 20, no. 11-13, pp. 1191–1199, 1999.

[10] J.-J. Huang, G.-H. Tzeng, and C.-S. Ong, "Marketing segmentation using support vector clustering," *Expert systems with applications*, vol. 32, no. 2, pp. 313–317, 2007.

[11] L. S. Oliveira and R. Sabourin, "Support vector machines for handwritten numerical string recognition," in *Ninth International Workshop on Frontiers in Handwriting Recognition*. IEEE, 2004, pp. 39–44.

[12] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy, "A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis," *Bioinformatics*, vol. 21, no. 5, pp. 631–643, 2004.

[13] V. N. Vapnik, "The nature of statistical learning," *Theory*, 1995.

[14] B. Aisen. A comparison of multiclass svm methods. [Online]. Available: https://courses.media.mit.edu/2006fall/mas622j/Projects/aisen-project/

[15] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin dags for multiclass classification," in *Advances in neural information processing systems*, 2000, pp. 547–553.

[16] B. Fei and J. Liu, "Binary tree of svm: a new fast multiclass training and classification algorithm," *IEEE transactions on neural networks*, vol. 17, no. 3, pp. 696–704, 2006.