SVM 的前世今生

1 Past

The origin of SVM can be dated back to 1947 when Frank Rosenblatt has just presented perceptron. Perceptron is a simple neural network and can classify data points into two parts. However, the perceptron a flaw: it requires the data to be linearly separable, meaning the data points can be perfectly separated by a straight line.

The basic idea of the perceptron are as follows: Suppose that in the $f : \mathbb{R}^n$ space, there have n discrete points, denoted as $\{x_1, x_2, \cdots x_n\}$ Each of them either have property1 or property2.

We want to draw a hyperplane to seperate these two types of points, that is to say: let one type of points in the one side of the hyperplane. We define $d_i = 1$ if x_i should above the hyperplane and $d_i = -1$ if x_i should below the hyperplane.

And we have $\boldsymbol{w} = [w_0, w_1, \cdots w_n]$ $\boldsymbol{x} = [1, coord_1, coord_2, \cdots coord_n]$

Then the hyperplane can be expressed as $l : \boldsymbol{w}^T\boldsymbol{x} = 0$ for each discrete point x_i, we substitute their coordinates into the hyperplane equation and we can get
$f(x_i) : \boldsymbol{w}^T\boldsymbol{x}_i > 0$ if x_i is actually above the hyperplane $f(x_i) : \boldsymbol{w}^T\boldsymbol{x_i} \leq 0$ if x_i is actually on the hyperplane or below the hyperplane If d_i and $f(x_i)$ have same sign, then x_i is classified corrected, else we need to adjust the hyperplane.

Then we can convert the problem to an optimization problem:
$minJ(\boldsymbol{w}) = \sum_{x_i \in M} -d_i \cdot dis(\boldsymbol{x_i})$
where M is the set of all wrong classified points and $dis(\boldsymbol{x_i})$ is the distance between pointx_i and the hyperplane. We can use gradient descent to solve this optimization problem.

Then, in 1963, Vapnik and his colleagues introduced SVM to address the limitations of the perceptron model. They pointed out "The Original Maximum Margin Hyperplane Algorithm". We have n data points: $(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \cdots, (\boldsymbol{x_n}, y_n)$ where $\boldsymbol{x_i}$ is n-dimension vector and y_i is 1 or -1,indicating the class of $\boldsymbol{x_i}$. If the data points are linearly separable, then we can find out two parallel hyperplanes that can seperate all points to their correct class and has the biggest distance between two hyperplanes. We name the two parallel hyperplanes " Maximum Margin Hyperplane". Then it is easy to see that all the parallel hyperplanes between Maximum Margin Hyperplane are solution to the classification problem.

The two parallel hyperplanes can be expressed as $\boldsymbol{w}\boldsymbol{x} + b = 1$ and $\boldsymbol{w}\boldsymbol{x} + b = -1$ The distance between is $\frac{2}{\|\boldsymbol{w}\|}$. We want to maximize the distance between the Maximum Margin

Hyperplane. And at the same time, we need to guarantee that for all i, it need to satisify one of the following two conditions:

$\boldsymbol{wx_i} + b \geq 1$ and $y_i = 1$

$\boldsymbol{wx_i} + b \leq 1$ and $y_i = -1$

The two condition can be expressed together as: $y_i(\boldsymbol{wx_i} + b) \geq 1$ for all i And further more, we can acknowledge that the Maximum Margin Hyperplane are decided absoluately from the closest $\boldsymbol{x_i}$, These $\boldsymbol{x_i}$ are called Support Vector.

In 1992, Bernhard E. Boser and Isabelle M. Guyon introduced the kernel function to map the unlinearly separable points to higher dimension space to make them linearly separable. That is to say, make the difference between different parts more significant. Some most used kernels:

$LinearKernel : K(x_i, x_j) = (x_i \cdot x_j)$

$PolynomialKernel : K(x_i, x_j) = (x_i \cdot x_j + 1)^p$

$GaussianKernel : K(x_i, x_j) = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}$

$RBFKernel : K(x_i, x_j) = e^{-\gamma(x_i - x_j)^2}$

$SigmoidKernel : K(x_i, x_j) = tanh(\eta x_i \cdot x_j + v)$

But It is note-worthing that a higher-dimensional feature space increases the generalization error of the support vector machine, but given enough samples, the algorithm can still perform well.

In 1993, Corinna Cortes introduced Soft Margin SVM. They introduced slack variable ξ_k and penalty factor C Then we want to: $y_i(\boldsymbol{wx_i} + b) \geq 1 - \xi_i$ for all i

and the optimization goal are set as: $min \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{k=1}^{k=n} \xi_k$

In 1990s, with rapid development of Machine learning, Vapnik intorduced Structural Risk Minimization(SRM) in 1970s get more and more attention. SRM is designed to find the best trade-off between a model's complexity and its ability to generalize to unseen data, helping to prevent overfitting.

When it comes to SRM, we have to mention Empirical Risk Minimization(ERM). ERM aims to minimize the error(lose) on training data, it focuses solely on training data. Thus ERM has a fatal flaw: it just focuses on fitting data as accurate as possible, without regard to model complexity, this will lead to overfitting – fit very good on training data but perform poorly on new, unseen data. The cause of overfit is model captures noise or irrelavant features.

SRM can alleviate overfitting by balancing a trade-off between the the model's fit on the training data and its complexity. Mathematically, SRM minimizes a form of generalization error, which is the sum of the training error and the of model complexity. The penalty grows with the complexity of the model, reflecting the risk of overfitting.

2 Present

We organize the development history of SVM we can conclude a comprehensive SVM:

Two set of data points:

$C1 = \{x_i \in \mathbb{R}^n : i = 1, 2, \cdots, m\}$, $C2 = \{x_i \in \mathbb{R}^n : i = m+1, m+2, \cdots, m+n\}$

Learn the hyperplane parameters $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ that:

$w^T x_i + b > 0, (i = 1, 2, \cdots, m)$

$w^T x_j + b < 0, (j = m+1, m+2, \cdots, m+n)$

Maximize the shortest distance from data point to hyperplane:

$\max_{w,b} min\{\|x - x_k\|_2 : w^T x + b = 0, k = 1, 2, \cdots, m+n\}$

That make the shortest distance data points on the two hyperplanes $w^T x_i + b = \pm 1$respectively, so that

$w^T x_i + b \geq 0, (i = 1, 2, \cdots, m)$

$w^T x_j + b \leq 0, (j = m+1, m+2, \cdots, m+n)$

The distance between two hyperplanes

$H1 = \{z : w^T z + b = 1\}$ and $H2 = \{z : w^T z + b = -1\}$ is $\frac{2}{\|\boldsymbol{w}\|}$

Then we can make the optimization problem:

$\min_{w \in \mathbb{R}^n, b \in \mathbb{R}} \frac{1}{2}\|w\|_2$

$s.t. w^T x_i + b \geq 1, (i = 1, 2, \cdots, m)$

$w^T x_j + b \leq -1, (j = m+1, m+2, \cdots, m+n)$

Positive label $y_i = 1, (i = 1, 2, \cdots, m)$

Negative label $y_j = -1, (j = m+1, m+2, \cdots, m+n)$

We can get:

$\min_{w \in \mathbb{R}^n, b \in \mathbb{R}} \frac{1}{2}\|w\|_2$

$s.t. y_i(w^T x_i + b) \geq 1, (i = 1, 2, \cdots, m+n)$

But what will happen if there does not eaxist (w, b) that can seperate data points into two parts?

We introduce slack variable ξ_k and penalty factor C, that there exists (w, b) that satisify $y_k(w^T x_k + b) \geq 1 - \xi_k, (i = 1, 2, \cdots, m+n)$

That the optimization problem can be converted into:

$\min_{w \in \mathbb{R}^n, b \in \mathbb{R}, \xi_k (\forall k)} \frac{1}{2}\|w\|_2 + C\sum_{k=1}^{k=n} \xi_k$

$y_k(w^T x_k + b) \geq 1 - \xi_k, (i = 1, 2, \cdots, m+n)$

$\xi_k \geq 0 (k = 1, 2, \cdot, m+n)$

There are three optimized variables: $w \in \mathbb{R}^n, \xi = [\xi_1, \xi_2, \cdots, \xi_m + n]^T \in \mathbb{R}^m + n, b \in \mathbb{R}$

The SVM optimization can actually be seen as a quadratic programming problem:

$$\text{x=}\begin{bmatrix} w \\ \xi \\ b \end{bmatrix},\text{P} = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\text{c} = \begin{bmatrix} 0 \\ C \cdot 1 \\ 0 \end{bmatrix},\text{G} = \begin{bmatrix} -diag(y)X & -I & -y \\ 0 & -I & 0 \end{bmatrix}$$

$$\text{y=}\begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_{m+n} \end{bmatrix}\ \text{X=}\begin{bmatrix} x_1^T \\ x_2^T \\ \cdots \\ x_{m+n} \end{bmatrix}\ \text{h=}\begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$min\frac{1}{2}x^TPx + c^Tx + d$
$s.t. Gx \leq h$
$\xi_k \geq 0$

Since that P is a symmetric positive semidefinite metrix and $Gx - h \leq 0$ is a convex function, then it is a convex optimization problem.
It's prove SVM optimization satisifies Slater condition: Let $w = 0, b = 0, \xi_k = 2$ then we have:
$y_k(w^Tx_k + b) = 0 > 1 - \xi_k = 1 - 2 = -1$
So the strong duality is satisified. So the SVM has the same optimal solution as its dual problem.
And since the inequality constraints of SVM original problem are too much and too complex, we can convert the SVM problem to its dual problem.
The dual problem: Lagrange function:
$L(w, b, \xi, \alpha, \beta) = \frac{1}{2}\|w\|_2^2 + C\sum_{k=1}^{m+n}\xi - \sum_{k=1}^{m+n}\alpha_k(y_k(w^Tx_k + b) - 1 + \xi_k) - \sum_{k=1}^{m+n}\beta_k\xi_k$
Lagrange dual function:
$g(\alpha, \beta) = \min_{w,b,\xi} L(w, b, \xi, \alpha, \beta)$
Lagrange dual problem: $\max_{\alpha\geq 0, \beta \geq 0} g(\alpha, \beta)$
To be more precise:
$g(\alpha, \beta) = \min_{w,b,\xi} L(w, b, \xi, \alpha, \beta) = \min_{w,b,\xi} \frac{1}{2}\|w\|_2^2 + C\sum_{k=1}^{m+n}\xi - \sum_{k=1}^{m+n}\alpha_k(y_k(w^Tx_k + b) - 1 + \xi_k) - \sum_{k=1}^{m+n}\beta_k\xi_k$
Since when we get optimal value of $\min_{w,b,\xi} L(w, b, \xi, \alpha, \beta)$, the partial difference for w, b, ξ are 0, then we have
$\frac{\partial L(w,b,\xi,\alpha,\beta)}{\partial w} = 0 \rightarrow w = \sum_{k=1}^{m+n}\alpha_k y_k x_k$
$\frac{\partial L(w,b,\xi,\alpha,\beta)}{\partial b} = 0 \rightarrow \sum_{k=1}^{m+n}\alpha_k y_k$

$\frac{\partial L(w,b,\xi,\alpha,\beta)}{\partial \xi} = 0 \rightarrow \alpha_k + \beta_k = C$

$g(\alpha,\beta) = \min_{\alpha} \sum_{k=1}^{m+n} \sum_{s=1}^{m+n} \alpha_k a_s y_k y_s x_i^T x_j - \sum_{k=1}^{m+n} \alpha_k$

$s.t. 0 \leq \alpha_k \leq C, i = 1, 2, 3, \cdots, m+n \sum_{k=1}^{m+n} \alpha_k y_k = 0$

That is the dual problem of SVM.

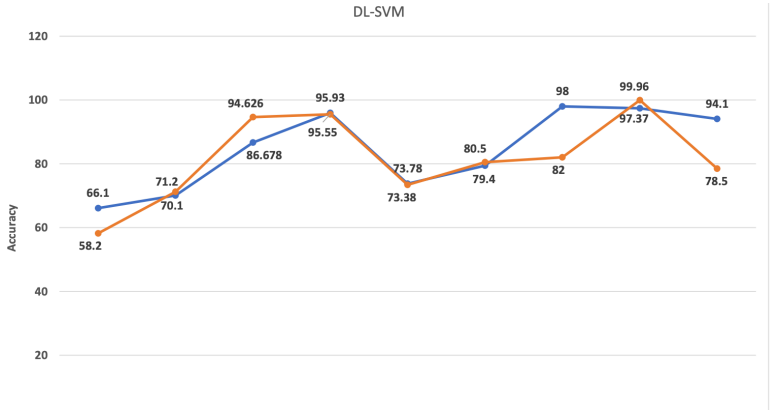
For its powerful classification capability, SVM is not just limited to easy classification tasks, it is widely used in various fields. For example: pattern recognition, Object recognition, image classification, text categorization, Bioinformatics etc.

At the same time,SVM has been integrated with other advanced methods such as evolve algorithms, to enhance the ability of classification and optimize parameters.

However, there are some limitations of SVM. First, the principal disadvantage of SVM is its huge computational cost in large data sets.(i.e. SVM is not suitable for large data set classification) At the same time, Support Vector Machines were originally designed to solve binary classification problems. When it comes to multiclassification, SVM does not present an easy solution. Though there are some ways to change the multiclassification problem to binary classification problem, the computational cost is very huge.

3 Future

Since There are limitations of SVM, we human beings always want to correct theses limitations. For multiclassification problems, Now there exists one-against-one and one-against-all approach. But each of them has their own drawback So ,for a large number of classes, new heuristic, stochastic or hybrid methods need to be designed to improve classification accuracy. And on thing that I am most interested in is Deep learning and SVM. In recent few years, deep learning has gain more and more attention and thus less attention are put on SVM.



Deep learning Vs SVM in some tasks(From [5])

Both of Deep learning and SVM has their own advantages and disadvantages. If we can let them work in synergy, I believe, the performance in some application will be improved.

Reference

[1]Support vector machine, Wikipedia, $https://en.wikipedia.org/wiki/Support_vector_machine$

[2] 致敬真神：SVM 的进化史, kyle, $https://www.bytezonex.com/archives/N0UqQYV4.html$

[3]Slides in "Fundamental Optimization" in Xi'an Jiaotong University, Minnan Luo

[4]Jair Cervantes,Farid Garcia-Lamont ,Lisbeth Rodríguez-Mazahua,Asdrubal Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends" Neurocomputing Volume 408, 30 September 2020, Pages 189-215

[5] M. A. Cano Lengua and E. A. Papa Quiroz, "A Systematic Literature Review on Support Vector Machines Applied to Classification," 2020 IEEE Engineering International Research Conference (EIRCON), Lima, Peru, 2020, pp. 1-4, doi10.1109/EIRCON51178.2020.9254028.