Optimization Course Project V:

Support Vector Machine

1 Introduction

Support vector machine (SVM) is a powerful tool for classification and regression in machine learning. The history of SVM can be tracked back to 1962, when it was first invented by Vladimir Vapnik and Alexey Chervonenkis in ([1]). Later, with the evolution of machine learning, coupled with numerous subsequent studies ([2, 3]), SVM became one fundamental machine learning tool, which enjoys both solid empirical effectiveness and strong theoretical interpretability. Its adaptability across a diverse array of domains, such as finance, healthcare, and manufacturing.

This project focuses on applying SVM to binary classification problems. Specifically, we consider two classes of data points. Denote $\{\mathbf{a}_i\}_{i=1}^n \subseteq \mathbb{R}^d$ and $\{\mathbf{b}_j\}_{j=1}^m \subseteq \mathbb{R}^d$ as training data from Class 1 and 2, respectively. In this case, SVM aims to identify a boundary between Classes 1 and 2 based on the training data $\{\mathbf{a}_i\}_{i=1}^n \subseteq \mathbb{R}^d$ and $\{\mathbf{b}_j\}_{j=1}^m \subseteq \mathbb{R}^d$, and then, it will classify any new point based solely on this boundary.

2 Different Variants of SVM

Here are some variants of SVM.

Hard-Margin SVM: We first consider a linear separable case, where there is a hyperplane such that $\{\mathbf{a}_i\}_{i=1}^n$ and $\{\mathbf{b}_j\}_{j=1}^m$ are situated on opposite sides of this hyperplane. To identify this hyperplane, hard-margin SVM suggests first selecting two parallel hyperplanes that separate the two classes of data by maximizing the distance between those two hyperplanes. Then, the hyperplane equidistant from the aforementioned parallel hyperplanes is selected as the boundary of the two classes. This idea can be formalized by the

following quadratic program. :

$$\begin{aligned} \min \quad & \|\mathbf{x}\|_2^2 \\ \text{subject to} \quad & \mathbf{a}_i^\top \mathbf{x} + x_0 \geq 1, \text{ for all } i = 1, \ldots, n, \\ & \mathbf{b}_j^\top \mathbf{x} + x_0 \leq -1, \text{ for all } j = 1, \ldots, m. \end{aligned} \tag{1}$$

Here, $\mathbf{x}$ and x_0 are the boundary hyperplane's slope vector and intersect scalar. In this case, then $\mathbf{w}^\top \mathbf{x} + x_0 = 1$ and $\mathbf{w}^\top \mathbf{x} + x_0 = -1$ are the two parallel hyperplanes mentioned before, and $\frac{2}{\|\mathbf{x}\|}$ is the distance between those two hyperplanes. In addition, We remark that the region between those two hyperplanes is also called the margin, the size of which then can be measured by $\frac{2}{\|\mathbf{x}\|}$, so the SVM classifier is also known as the maximum-margin classifier.

Soft-Margin SVM: A notable limitation of the hard-margin SVM arises when the data points from two classes are not strictly linearly separable. Specifically, the problem defined in (1) becomes infeasible even when the majority of data points from both sets lie on their respective sides of a hyperplane, except for several outliers due to random noises. To address this issue, [3] introduce some slack variables ξ_i and ζ_j to constraints in (1). Then, the new optimization problem can be reformulated as follows:

$$\begin{aligned} \min \quad & \|\mathbf{x}\|_2^2 + \mu \left(\sum_{i=1}^{n} \xi_i + \sum_{j=1^m} \zeta_j \right) \\ \text{subject to} \quad & \mathbf{a}_i^\top \mathbf{x} + x_0 \geq 1 - \xi_i, \text{ for all } i = 1, \ldots, n, \\ & \mathbf{b}_j^\top \mathbf{x} + x_0 \leq -1 + \zeta_j, \text{ for all } j = 1, \ldots, m \\ & \xi_i, \zeta_j \geq 0, \text{ for all } i, j, \end{aligned} \tag{2}$$

where $\mu > 0$ is a pre-fixed constant that potentially depends on d, n, and m. Here, the slack variables in the objective quantify the classification error of the two parallel hyperplanes. Compared to the hard-margin formulation (1), (2) also looks for two parallel hyperplanes with maximum margin to separate points in two classes, while it allows the margin, or the region between these two hyperplanes, to contain some data points. Thus, this formulation is known as soft-margin SVM.

SVM with Non-Linear Mapping: In the previous two parts, we assume the two classes are linearly separable. In this part, we discuss applying SVM if linear separability does not hold. For the sake of simplicity, we first consider an ellipsoidal separation setting, where $\{\mathbf{a}_i\}_{i=1}^n$ are almost out of an ellipsoid, and $\{\mathbf{b}_j\}_{j=1}^m$ are almost contained in the same ellipsoid. In this case, one can still apply SVM by mapping $\mathbf{a}_i$'s and $\mathbf{b}_j$'s to a linear separable space. Specifically, since there exists an ellipsoid separating $\{\mathbf{a}_i\}_{i=1}^n$ and

$\{\mathbf{b}_j\}_{j=1}^n$, one can find $\mathbf{X} \in \mathbb{R}_+^{d\times d}$ and $\mathbf{x} \in \mathbb{R}^d$ such that

$$\begin{aligned}\mathbf{a}_i\mathbf{a}_i^\top \cdot \mathbf{X} + \mathbf{a}_i^\top \mathbf{x} + x_0 > 0, \text{ for a majority of } i = 1, \ldots, n,\\ \mathbf{b}_j\mathbf{b}_j^\top \cdot \mathbf{X} + \mathbf{b}_j^\top \mathbf{x} + x_0 < 0, \text{ for a majority of } j = 1, \ldots, m,\end{aligned}$$

where $-\frac{\mathbf{x}}{2}$ is the center of the ellipsoid, and $\mathbf{X}$ measures the size and direction of the ellipsoid. Then, letting

$$\phi(\mathbf{a}) = (\mathbf{a}\mathbf{a}^\top, \mathbf{a}) \in \mathbb{R}^{d\times d} \times \mathbb{R}^d$$

for $\mathbf{a} \in \mathbb{R}^d$, we have that $\{\phi(\mathbf{a}_i)\}_{i=1}^n$ and $\{\phi(\mathbf{b}_j)\}_{j=1}^m$ are linearly separable, and thus, we can apply soft-margin SVM by solving the following mixed linear and semidefinite programming problem:

$$\begin{aligned}\min \quad & \text{Trace}(\mathbf{X}) + \|\mathbf{x}\|_2^2 + \mu\left(\sum_{i=1}^n \xi_i + \sum_{j=1^m} \zeta_j\right)\\ \text{subject to} \quad & \mathbf{a}_i\mathbf{a}_i^\top \cdot \mathbf{X} + \mathbf{a}_i^\top \mathbf{x} + 1 \cdot x_0 \geq 1 - \xi_i, \text{ for all } i = 1, \ldots, n,\\ & \mathbf{b}_j\mathbf{b}_j^\top \cdot \mathbf{X} + \mathbf{b}_j^\top \mathbf{x} + 1 \cdot x_0 \leq -1 + \zeta_j,\\ & \xi_i, \zeta_j \geq 0, \text{ for all } j = 1, \ldots, m.\end{aligned} \tag{3}$$

This idea of SVM with non-linear mapping also works in general. Particularly, for any mapping ϕ, one can still construct a soft-margin SVM problem as follows:

$$\begin{aligned}\min \quad & \|\mathbf{x}\|_2^2 + \mu\left(\sum_{i=1}^n \xi_i + \sum_{j=1^m} \zeta_j\right)\\ \text{subject to} \quad & \phi(\mathbf{a}_i)^\top \mathbf{x} + x_0 \geq 1 - \xi_i, \text{ for all } i = 1, \ldots, n,\\ & \phi(\mathbf{b}_j)^\top \mathbf{x} + x_0 \leq -1 + \zeta_j, \text{ for all } j = 1, \ldots, m,\\ & \xi_i, \zeta_j \geq 0, \text{ for all } i, j.\end{aligned} \tag{4}$$

Kernalized SVM (Optional): To further generalize SVM with non-linear mapping, [2] introduces kernelized SVM, which we will discuss below. This kernelized SVM can generalize the above variants of SVM without explicitly bothering the mapping ϕ, and make SVM more powerful in practice.

Here, we derive the kernelized SVM. Specifically, denote $(\mathbf{x}^*, x_0^*)$ as the optimal solution of (2). Then, based on KKT conditions of (2), one can show

$$\begin{aligned}\mathbf{x}^* &= \sum_{i=1}^n \alpha_i \phi(\mathbf{a}_i) - \sum_{j=1}^m \beta_j \phi(\mathbf{b}_j),\\ x_0^* &= \arg\min \sum_{i=1}^n (1 - x_0 - \phi(\mathbf{a}_i)^\top \mathbf{x}) + \sum_{j=1}^m (1 + x_0 + \phi(\mathbf{b}_j)^\top \mathbf{x}).\end{aligned} \tag{5}$$

In the above equalities, α_i's and β_j's are the solutions of the dual problem of (2) as listed below (please double-check it by yourself).

$$\begin{aligned} \max \quad & \sum_{i=1}^{n}\alpha_i + \sum_{j=1}^{m}\beta_j + \sum_{i_1,i_2=1}^{n}\alpha_{i_1}\alpha_{i_2}\phi(\mathbf{a}_{i_1})^\top\phi(\mathbf{a}_{i_2}) + \sum_{j_1,j_2=1}^{m}\beta_{j_1}\beta_{j_2}\phi(\mathbf{b}_{j_1})^\top\phi(\mathbf{b}_{j_2}) - \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i\beta_j\phi(\mathbf{a}_i)^\top\phi(\mathbf{b}_j) \\ \text{subject to} \quad & \sum_{i=1}^{n}\alpha_i = \sum_{j=1}^{m}\beta_j \\ & 0 \le \alpha_i, \beta_j \le \mu \text{ for all } i = 1,\dots,n,\ j = 1,\dots,m \end{aligned} \tag{6}$$

Then, for any new point $\mathbf{c}$, one can classify it by the following formula,

$$\phi(\mathbf{c})^\top\mathbf{x}^* + x_0 \begin{cases} > 0, & \mathbf{c} \in \text{Class 1}, \\ < 0, & \mathbf{c} \in \text{Class 2}. \end{cases} \tag{7}$$

Now, we define the kernel $K(\cdot,\cdot) := \phi(\cdot)^\top\phi(\cdot)$ be the inner product of the mapped points. With the kernel, we can simplify the classification rule (7) as follows:

$$\sum_{i=1}^{n}\alpha_i K(\mathbf{a}_i,\mathbf{c}) - \sum_{j=1}^{m}\beta_j K(\mathbf{b}_j,\mathbf{c}) \begin{cases} > 0, & \mathbf{c} \in \text{Class 1}, \\ < 0, & \mathbf{c} \in \text{Class 2}. \end{cases} \tag{8}$$

We can see that (8) depends only on the kernel K since x_0 also depends only on the kernel. In addition, with this kernel, we can rewrite (6) to (9)

$$\begin{aligned} \max \quad & \sum_{i=1}^{n}\alpha_i + \sum_{j=1}^{m}\beta_j + \sum_{i_1,i_2=1}^{n}\alpha_{i_1}\alpha_{i_2}K(\mathbf{a}_{i_1},\mathbf{a}_{i_2}) + \sum_{j_1,j_2=1}^{m}\beta_{j_1}\beta_{j_2}K(\mathbf{b}_{j_1},\mathbf{b}_{j_2}) - \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i\beta_j K(\mathbf{a}_i,\mathbf{b}_j) \\ \text{subject to} \quad & \sum_{i=1}^{n}\alpha_i = \sum_{j=1}^{m}\beta_j, \\ & 0 \le \alpha_i, \beta_j \le \mu \text{ for all } i = 1,\dots,n, j = 1,\dots,m. \end{aligned} \tag{9}$$

This new optimization problem also depends only on the kernel K. Combining (8) and (9), if the kernel K is given, one can directly find the classifier (8) without explicitly using the mapping ϕ. Actually, Mercer's Theorem says that there exists a mapping ϕ such that $K(\cdot,\cdot) = \phi(\cdot)^\top\phi(\cdot)$ if and only if a function K satisfies i) nonnegativity $K(\mathbf{a},\mathbf{b}) \ge 0$, and ii) symmetry: $K(\mathbf{a},\mathbf{b}) = K(\mathbf{b},\mathbf{a})$ for all $\mathbf{a},\mathbf{b}$. Thus, in practice, one can apply any kernels to find a classifier by (8) and (9) without constructing a feature mapping ϕ. This method is also known as the "kernel method." Some popular kernel functions are listed below.

$$\begin{aligned} \text{Linear Kernel: } & K(\mathbf{a},\mathbf{b}) = \mathbf{a}^\top\mathbf{b}, \\ \text{Gaussian Kernel: } & K(\mathbf{a},\mathbf{b}) = \exp\{-\gamma\|\mathbf{a}-\mathbf{b}\|_2^2\}, \text{ for some parameter } \gamma > 0, \\ \text{polynomial Kernel: } & K(\mathbf{a},\mathbf{b}) = (\mathbf{a}^\top\mathbf{b} + r)^d, \text{ for some constant } r \text{ and degree } d. \end{aligned}$$

3 Project Goals

You may explore those SVM variants by applying SVMs on some generated linear or ellipsoidal separable binary classification problems and on the MNIST Dataset. The key questions are: what are the differences among those different variants? What are the best choices of μ? Are SVMs robust to some outliers (or extreme points) of the training samples? How are the performances of SVMs if the data are not linear and ellipsoidal separable? How can you accelerate SVM's training process for large-scale problems? For the kernelized SVM, which kernel is the best? How is the performance of kernels that do not satisfy the nonnegativity condition? The comparison can include aspects such as algorithm design, theoretical analysis, computation time, and the approximation error of different algorithms.

4 Remarks

In recent years, SVM might not have garnered as much attention as other areas within the machine learning community. However, SVM's empirical efficacy and theoretical clarity continue to underscore its significance as a practical tool. Moreover, the concepts of maximum margin and the kernel method also have considerable influence in the machine learning community, including but not limited to applying and understanding deep neural networks ([4, 5]).

References

[1] Chervonenkis, Alexey Ya Early history of support vector machines Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik, 2013

[2] Boser, Bernhard E and Guyon, Isabelle M and Vapnik, Vladimir N A training algorithm for optimal margin classifiers Proceedings of the fifth annual workshop on Computational learning theory, 1992

[3] Corinnna Cortes and Vladimir Vapnik, Support-Vector Networks Machine Learning, 1995

[4] Elsayed, Gamaleldin and Krishnan, Dilip and Mobahi, Hossein and Regan, Kevin and Bengio, Samy Large margin deep networks for classification Advances in neural information processing systems, 2018

[5] Golikov, Eugene and Pokonechnyy, Eduard and Korviakov, Vladimir Neural tangent kernel: A survey arXiv preprint arXiv:2208.13614