Data-dependent Kernel Support Vector Machine classifiers in Reproducing Kernel Hilbert Space

Hyemin Gu[1]

[1]University of Massachusetts Amherst
hgu@umass.edu

Abstract

Support Vector Machine (SVM) provides a linear classifier for binary classification problems. Complex decision boundaries in the input feature space are handled by nonlinear kernels to the SVM. Theories in Reproducing Kernel Hilbert Spaces (RKHS) state that, given a kernel $\mathcal{K}$ and a set of M given data $\{x_i, y_i\}_{i=1}^M$, a SVM classifier function can be written as $f(x) = \alpha_0 + \sum_{i=1}^M \alpha_i \mathcal{K}(x, x_i)$ for some coefficients α_is. Also, applying conformal transforms to a positive definite kernel produces another positive definite kernel which are in more complexity. Hence, in case that well-known kernels fail given the current training data, a new kernel can be tried by optimizing the coefficients of a conformal kernel in the way to maximize the ratio "(Between-class error)/(Within-class error)" of the training data. Here, data-dependent kernel SVM is applied to an application of classifying tumor/tumor-free organs from gene expression data and compared its classification performance with other well-known kernels.

Keywords: Data-dependent, Support Vector Machine, Reproducing Kernel Hilbert Space

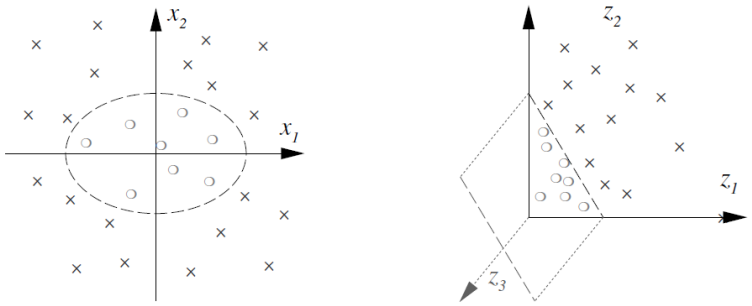
Contents

1 Problem statement

Consider a *binary classification problem* which is stated as below.

Let X be an input space and $Y = \{\pm 1\}$ be an output space for two classes. Given paired data $\{(x_i, y_i)\}_{i=1,\cdots,N} \subset X \times Y$, build a classifier $T : X \to Y$. It is considered that T separates the input space X into several regions providing decision boundaries that separate the inputs.

Depending on the structure of inputs, decision boundaries could be linear or nonlinear.



Approaches to find the decision boundary induce optimization problems. Support vector machine is one of them.

2 Mathematical Preliminaries

2.1 Support Vector Machines

The formulation and notations are following [2].

2.1.1 Formulation of the optimization problem

Let us first consider the case that a hyperplane $\{x : f(x) = x^T\beta + \beta_0 = 0\}$ where $\|\beta\| = 1$ clearly separates the classes. One idea to build such a hyperplane is to maximize the margin M between points and the plane. Since the classes are separable, we can find a function f such that $y_i f(x_i) > 0 \ \forall i$. This is called a Support vector machine (SVM) and can be witten as

$$\max_{\beta,\beta_0,\|\beta\|=1} M \tag{1}$$

subject to $y_i(x_i^T\beta + \beta_0) \geq M, \ i = 1, \cdots, N$.

The assumption that $\|\beta\| = 1$ can be relaxed by setting $M = 1/\|\beta\|$, so the problem can be reformulated as

$$\min_{\beta,\beta_0} \|\beta\| \tag{2}$$

subject to $y_i(x_i^T\beta + \beta_0) \geq 1, \ i = 1, \cdots, N$. This is a convex optimization problem with quadratic criterion.

Now assume that the classes overlap in the input space. Define the slack variable $\xi = (\xi_1, \xi_2, \cdots, \xi_N)$ where $\xi_i \geq 0$, $\sum_{i=1}^N \xi_i \leq$ constant to indicate the overlaps. Retaining the convexity of the problem we can formulate a similar problem.

$$\min_{\beta,\beta_0} \|\beta\| \tag{3}$$

subject to $y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i, \ i = 1, \cdots, N, \xi_i \geq 0, \sum_{i=1}^N \xi_i \leq$ constant. This is referred to the *standard support vector classifier*. Note that misclassification occurs if $\xi_i > 1$.

2.1.2 Lagrangian method and its solution

Using Lagrangian multipliers, the problem (3) with slack variables can be rewritten as a quadratic programming

$$\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^N \xi_i \tag{4}$$

subject to $y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i,\ i = 1, \cdots, N, \xi_i \geq 0$ where C is a parameter.

Lagrangian primal function is

$$L_P = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(x_i^T\beta + \beta_0) - (1 - \xi_i)] - \sum_{i=1}^{N}\mu_i\xi_i \tag{5}$$

where $\alpha_i, \mu_i \geq 0$. $\min_{\beta,\beta_0,\xi_i} L_P$ can be obtained when

$$\beta = \sum_{i=1}^{N}\alpha_i y_i x_i, \tag{6}$$

$$0 = \sum_{i=1}^{N}\alpha_i y_i, \tag{7}$$

$$\alpha_i = C - \mu_i, \tag{8}$$

$\forall i$. Therefore, we can get Lagrangian dual function by plugging them in the primal function L_P.

$$L_D = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}^{N}\alpha_i\alpha_{i'}y_i y_{i'}x_i^T x_{i'} \tag{9}$$

subject to $0 \leq \alpha_i \leq C$ and $0 = \sum_{i=1}^{N}\alpha_i y_i$. Additional constraints

$$\alpha_i[y_i(x_i^T\beta + \beta_0) - (1 - \xi_i)] = 0, \tag{10}$$

$$\mu_i\xi_i = 0, \tag{11}$$

$$y_i(x_i^T\beta + \beta_0) - (1 - \xi_i) \geq 0 \tag{12}$$

for $i = 1, \cdots, N$ should be included to apply KKT condition. Then, by KKT condition, the optimal point for $\max_{\alpha_i} L_D$ is necessarily the solution to the primal and dual problem.

According to (6), the solution for β has the form

$$\hat{\beta} = \sum_{i=1}^{N}\hat{\alpha}_i y_i x_i \tag{13}$$

with nonzero $\hat{\alpha}_i$ only for those observations i for which the constraint in (12) are exactly met (due to (10)). Those observations are called the *support vectors*. For these support vectors, $\hat{\xi}_i = 0$ and due to (8) and (11), $0 < \hat{\alpha}_i < C$ and otherwise, $\hat{\alpha}_i = 0$. Such margins ($0 < \hat{\alpha}_i < C$ and $\hat{\xi}_i = 0$) are used to determine $\hat{\beta}_0$.

Given the solutions $\hat{\beta}$ and $\hat{\beta}_0$, the decision function T can be found as $\hat{T} = sign[\hat{f}(x)]$.

2.1.3 Kernel SVM

So far, the SVM classifier finds linear boundaries in the input space X. Generally, linear boundaris in the enlarged space achieve better separation, and translate to nonlinear boundaries in the original space. Transform the input features as $h(x_i) = (h_1(x_i), h_2(x_i), \cdots, h_M(x_i)) \in \mathbb{R}^M$ and produce nonlinear function $\hat{f}(x) = h(x)^T\hat{\beta} + \hat{\beta}_0$ and the classifier $\hat{T} = sign[\hat{f}(x)]$. In this way, the enlarged input space could be very large, so that it is computationally prohibitive. Recall that the Lagrange dual function (9) has the form

$$L_D = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}^{N}\alpha_i\alpha_{i'}y_i y_{i'} < h(x_i), h(x_{i'}) > \tag{14}$$

and from (6), the solution function $f(x)$ has the form

$$f(x) = h(x)^T\beta + \beta_0 \tag{15}$$

$$= \sum_{i=1}^{N}\alpha_i y_i < h(x), h(x_i) > +\beta_0. \tag{16}$$

Therefore, we require only the knowledge of the kernel function $K(x, x') =< h(x), h(x') >$ instead of the transformation $h(x)$. Particular choices of h give us cheaper calculations of the kernel $K(x, x')$ such as

$$\text{dth-degree polynomial: } K(x, x') = (1+ < x, x' >)^d, \tag{17}$$

$$\text{radial basis: } K(x, x') = exp(\gamma\|x - x'\|^2), \tag{18}$$

$$\text{sigmoid: } K(x, x') = tanh(\kappa_1 < x, x' > +\kappa_2). \tag{19}$$

2.2 Reproducing kernel Hilbert spaces

We are going to optimize above problems over certain function spaces. Hilbert spaces defined by reproducing kernels (RKHS) have computationally attractive properties. Further details are in [4].

2.2.1 Definition of RKHS

Note Hilbert space $\mathbb{H}$ is an inner product space with inner product $< \cdot, \cdot >$ which is complete. (i.e. Every Cauchy sequence converges in $\mathbb{H}$.) Examples are $\mathbb{R}^d$, L_2, ℓ_2, etc.

A linear functional on a Hilbert space $\mathbb{H}$ is a mapping $L : \mathbb{H} \to \mathbb{R}$ that is linear. And a linear functional L is bounded if for all $f \in \mathbb{H}$, there exists $M \leq \infty$ such that $|L(f)| \leq M\|f\|_{\mathbb{H}}$. The Riesz representation theorem characterizes bounded linear functionals in a Hilbert space.

Riesz representation theorem Let L be a bounded linear functional on a Hilbert space. Then there exists a unique $g \in \mathcal{H}$ such that $L(f) =< f, g >_{\mathbb{H}}$ for all $f \in \mathbb{H}$. (g is referred to the representer of the functional L.)

According to Riesz representation theorem, a bounded linear functionals in a Hilbert space is an inner product of a representer. In RKHS, we will see that the kernel acts as the representer for the evaluation functional.

A symmetric bivirate function $\mathcal{K} : X \times X \to \mathbb{R}$ is *positive semidefinite* if for all $n \in \mathbb{N}$ and elements $\{x_i\}_{i=1,\cdots,n} \subset X$, the $n \times n$ matrix with elements $\mathcal{K}_{ij} = \mathcal{K}(x_i, x_j)$ is positive semidefinite. By defining a mapping $\Phi : X \to Y$ where both X and Y are Hilbert spaces, $\mathcal{K}$ can be expressed as a *Gram matrix* of the form $\mathcal{K}(x, z) =< \Phi(x), \Phi(z) >_Y$.

Any PSD kernel $\mathcal{K}$ can be used to construct a particular and unique Hilbert space of functions. And this Hilbert space is unique and has the *kernel reproducing property*

$$< f, \mathcal{K}(\cdot, x) >_{\mathbb{H}} = f(x) \ \forall f \in \mathbb{H}. \tag{20}$$

It allows to define a feature map $x \mapsto \mathcal{K}(\cdot, x) \in \mathbb{H}$ from the kernel $\mathcal{K}$. The reproducing property ensures that

$$< \mathcal{K}(\cdot, x), \mathcal{K}(\cdot, z) >_{\mathbb{H}} = \mathcal{K}(x, z) \ \forall x, z \in \mathbb{X}. \tag{21}$$

To define a Hilbert space with the reproducing property (20), begin with a set $\tilde{\mathbb{H}}$ of functions of the form $f(\cdot) = \sum_{j=1}^{n} \alpha_j \mathcal{K}(\cdot, x_j)$ for some integer $n \geq 1$, set of points $\{x_j\}_{j=1}^{n} \subset X$ and weight vector $\alpha \in \mathbb{R}^n$. It can be shown that $\tilde{\mathbb{H}}$ is a vector space with inner product of $f(\cdot) = \sum_{j=1}^{n} \alpha_j \mathcal{K}(\cdot, x_j)$ and $g(\cdot) = \sum_{k=1}^{m} \beta_k \mathcal{K}(\cdot, x_k)$ defined as

$$< f, g >_{\tilde{\mathbb{H}}} := \sum_{j=1}^{n} \sum_{k=1}^{m} \alpha_j \beta_k \mathcal{K}(x_j, y_k). \tag{22}$$

Moreover, this inner product satisfy the reproducing property (20),

$$< f, \mathcal{K}(\cdot, x) >_{\tilde{\mathbb{H}}} := \sum_{j=1}^{n} \alpha_j \mathcal{K}(x_j, x) = f(x) \tag{23}$$

where the kernel acts as the representer for the evaluation functional.

Finally, given any Hilbert space $\mathbb{H}$ in which the evaluation functionals are bounded, there is a unique PSD kernel $\mathcal{K}$ that satisfies the reproducing property (20).

2.2.2 Mercer's theorem

Mercer's theorem states the decomposition of a kernel to PSD kernels.

For a nonnegative measure $\mathbb{P}$ over a compact metric space X, consider the function class $L^2(X;\mathbb{P})$ or simply $L^2(X)$ with the norm

$$\|f\|^2_{L^2(X;\mathbb{P})} = \int_X |f(x)|^2 d\mathbb{P}(x). \tag{24}$$

Given a symmetric PSD kernel $\mathcal{K} : X \times X \to \mathbb{R}$ which is continuous and $\int_{X\times X} \mathcal{K}^2(x,z)d\mathbb{P}(x)d\mathbb{P}(z) < \infty$, define a linear operator $T_\mathcal{K}$ on $L^2(X)$ via

$$T_\mathcal{K}(f)(x) := \int_X \mathcal{K}(x,z)f(z)d\mathbb{P}(z). \tag{25}$$

Apply Cauchy-Schwartz inequality

$$\|T_\mathcal{K}(f)\|^2_{L^2(X)} = \int_X \left(\int_X \mathcal{K}(x,z)f(x)d\mathbb{P}(x)\right)^2 d\mathbb{P}(z) \tag{26}$$

$$\leq \|f\|^2_{L^2(X)} \int_{X\times X} \mathcal{K}^2(x,z)d\mathbb{P}(x)d\mathbb{P}(z) \tag{27}$$

to show that $T_\mathcal{K}$ is a bounded operator on $L^2(X)$. Operators of these type are known as *Hilbert-Schmidt operators*.

Mercer's theorem Suppose that X is compact, the kernel $\mathcal{K}$ is continuous and PSD, and satisfies the Hilbert-Schmidt condition (26). Then there exists a sequence of eigenfunctions $(\phi_j)_{j=1}^\infty$ that form an orthonormal basis of $L^2(X;\mathbb{P})$, and its corresponding nonnegative eigenvalues $(\mu_j)_{j=1}^\infty$ such that

$$T_\mathcal{K}(\phi_j) = \mu_j\phi_j \text{ for } j = 1, 2, \cdots. \tag{28}$$

Moreover, the kernel function has the expansion

$$\mathcal{K}(x,z) = \sum_{j=1}^\infty \mu_j\phi_j(x)\phi_j(z), \tag{29}$$

where the convergence of the series holds absolutely and uniformly.

Mercer's theorem induces an idea on RKHS to provide an embedding of the function domain X into a subset of the sequence space $\ell^2(\mathbb{N})$. Using the eigenfunctions and eigenvalues from Mercer's theorem, define a mapping $\Phi : X \to \ell^2(\mathbb{N})$ via

$$x \mapsto \Phi(x) := (\sqrt{\mu_1}\phi_1(x), \sqrt{\mu_2}\phi_2(x), \sqrt{\mu_3}\phi_3(x), \cdots). \tag{30}$$

By construction,

$$\|\phi_1(x)\|^2_{\ell^2(\mathbb{N})} = \sum_{j=1}^\infty \mu_j\phi_j^2(x) = \mathcal{K}(x,x) < \infty, \tag{31}$$

and

$$< \phi_1(x), \phi_1(z) >^2_{\ell^2(\mathbb{N})} = \sum_{j=1}^\infty \mu_j\phi_j(x)\phi_j(z) = \mathcal{K}(x,z). \tag{32}$$

Corollary of Mercer's theorem Consider a kernel satisfying the conditions of Mercer's theorem with associated eigenfunctions $(\phi_j)_{j=1}^\infty$ and nonnegative eigenvalues $(\mu_j)_{j=1}^\infty$. It induces the RKHS

$$\mathbb{H} := \left\{ f = \sum_{j=1}^\infty \beta_j\phi_j(x) | (\mu_j)_{j=1}^\infty \subset \ell^2(\mathbb{N}), \sum_{j=1}^\infty \frac{\beta_j^2}{\mu_j} < \infty \right\}, \tag{33}$$

along with inner product

$$< f, g >_\mathbb{H} := \sum_{j=1}^\infty \frac{< f, \phi_j >< g, \phi_j >}{\mu_j} < \infty, \tag{34}$$

where $< \cdot, \cdot >$ denotes the inner product in $L^2(X;\mathbb{P})$.

2.2.3 operations on RKHS

Now we will see a number of operations on RKHS that allow us to build new spaces which are referred in [3].

First, define two Hilbert spaces from given Hilbert spaces $\mathbb{H}_1$ and $\mathbb{H}_2$ of functions defined on domains X_1 and X_2, respectively. Consider two spaces

- Addition:
$$\mathbb{H}_1 + \mathbb{H}_2 := \{f_1 + f_2 | f_j \in \mathbb{H}_j, j = 1, 2\} \tag{35}$$
with norm $\|f\|_{\mathbb{H}^2} := \min_{f=f_1+f_2, f_1\in\mathbb{H}_1, f_2\in\mathbb{H}_2}\{\|f\|^2_{\mathbb{H}_1} + \|f\|^2_{\mathbb{H}_2}\}$

- Tensor product:
$$\mathbb{H}_1 \otimes \mathbb{H}_2 := \{h : X_1 \times X_2 \to \mathbb{R} | h = \sum_{j=1}^{n} f_j g_j \text{ for some } n \in \mathbb{N}, f_j \in \mathbb{H}_1, g_j \in \mathbb{H}_2 \ \forall j \in \mathbb{N}\} \tag{36}$$
where its inner product is defined for $h = \sum_{j=1}^{n} f_j g_j$ and $\tilde{h} = \sum_{j=1}^{m} \tilde{f}_j \tilde{g}_j$ as $< h, \tilde{h} >_{\mathbb{H}} := \sum_{j=1}^{n}\sum_{k=1}^{m} < f_j, \tilde{f}_k >_{\mathbb{H}_1} < g_j, \tilde{g}_k >_{\mathbb{H}_2}$.

Now, the operations below defines a kernel from given kernels.

- sums and limits of kernels
The set of kernels forms a convex cone, closed under pointwise convergence.
  - If $\mathcal{K}_1$ and $\mathcal{K}_2$ are kernels, and $\alpha_1, \alpha_2 \geq 0$, then $\alpha_1\mathcal{K}_1 + \alpha_2\mathcal{K}_2$ is a kernel. Moreover, the kernel
  $$\mathcal{K} = \mathcal{K}_1 + \mathcal{K}_2 \tag{37}$$
  reproduces the RKHS in (35).
  - If $\mathcal{K}_1, \mathcal{K}_2, \cdots$ are kernels, and $\mathcal{K}(x, z) := \lim_{n\to\infty} \mathcal{K}_n(x, z)$ exists for all x, z, then $\mathcal{K}$ is a kernel.

- pointwise products of kernels
If $\mathcal{K}_1, \mathcal{K}_2$ are kernels, then $\mathcal{K}_1\mathcal{K}_2$, defined by
$$(\mathcal{K}_1\mathcal{K}_2)(x, z) := \mathcal{K}_1(x, z)\mathcal{K}_2(x, z) \tag{38}$$
is a kernel.
  - A special case is conformal transforms,
  $$\mathcal{K}_f(x, z) = f(x)\mathcal{K}(x, z)f(z) \tag{39}$$
  obtained by multiplying a kernel $\mathcal{K}$ with a rank-one kernel $\mathcal{K}' = f(x)f(z)$ where f is a positive function. Since
  $$\cos\left(\angle\left(\Phi_f(x), \Phi_f(z)\right)\right) = \frac{f(x)\mathcal{K}(x, z)f(z)}{\sqrt{f(x)\mathcal{K}(x, x)f(x)}\sqrt{f(z)\mathcal{K}(z, z)f(z)}} \tag{40}$$
  $$= \frac{\mathcal{K}(x, z)}{\sqrt{\mathcal{K}(x, x)}\sqrt{\mathcal{K}(z, z)}} = \cos\left(\angle\left(\Phi(x), \Phi(z)\right)\right), \tag{41}$$
  this transform does not affect the angles in the feature spaces.

- Dot product kernels
A differentiable function of the dot product $\mathcal{K}(x, z) = \mathcal{K}(< x, z >)$ has to satisfy
$$\mathcal{K}(t) \geq 0, \mathcal{K}'(t) \geq 0, \mathcal{K}'(t) + t\mathcal{K}''(t) \geq 0 \tag{42}$$
for any $t \geq 0$, in order to be a PSD kernel. A function $\mathcal{K}(x, z) = \mathcal{K}(< x, z >)$ defined on an infinite dimensional Hilbert space, with a power series expansion
$$\mathcal{K}(t) = \sum_{n=0}^{\infty} a_n t^n, \tag{43}$$
is a PSD kernel iff for all n, we have $a_n \geq 0$. A slightly weaker condition applies for finite dimensional spaces.

- tensor product kernels
  If $\mathcal{K}_1, \mathcal{K}_2$ are kernels defined respectively on $X_1 \times X_1$ and $X_2 \times X_2$, then their tensor product,

$$(\mathcal{K}_1 \otimes \mathcal{K}_2)(x_1, x_2, z_1, z_2) = \mathcal{K}_1(x_1, z_1)\mathcal{K}(x_2, z_2), \tag{44}$$

  is a kernel on $(X_1 \times X_2) \times (X_1 \times X_2)$ where $x_1, z_1 \in X_1$ and $x_2, z_2 \in X_2$. Moreover, the kernel $\mathcal{K}_1 \otimes \mathcal{K}_2$ reproduces the RKHS in (36).

- direct sums
  If $\mathcal{K}_1, \mathcal{K}_2$ are kernels defined respectively on $X_1 \times X_1$ and $X_2 \times X_2$, then their direct sum,

$$(\mathcal{K}_1 \oplus \mathcal{K}_2)(x_1, x_2, z_1, z_2) = \mathcal{K}_1(x_1, z_1) + \mathcal{K}(x_2, z_2), \tag{45}$$

  is a kernel on $(X_1 \times X_2) \times (X_1 \times X_2)$ where $x_1, z_1 \in X_1$ and $x_2, z_2 \in X_2$.

2.3 SVM classifiers in RKHS

Suppose that the transformed feature h arises from the eigen-expansion of a positive definite kernel $\mathcal{K}$,

$$\mathcal{K}(x, x') = \sum_{i=1}^{\infty} \phi_m(x)\phi_m(x')\delta_m \tag{46}$$

and $h_m(x) = \sqrt{\delta_m}\phi_m(x)$. Then this kernel reproduces a RKHS $\mathbb{H}_\mathcal{K}$ where functions in the $\mathbb{H}_\mathcal{K}$ are of the form

$$f(x) = \sum_{i=1}^{\infty} c_i \phi_i(x) \tag{47}$$

with the constraint that

$$\|f(x)\|^2_{\mathbb{H}_\mathcal{K}} := \sum_{i=1}^{\infty} \frac{c_i^2}{\delta_i} < \infty. \tag{48}$$

Letting a penalty functional $J(f) = \|f(x)\|^2_{\mathbb{H}_\mathcal{K}}$, consider the regularized optimization problems which are equivalent

$$\min_{f \in \mathbb{H}_\mathcal{K}} \left[\sum_{i=1}^{N} L(y_i, f(x_i)) + \frac{\lambda}{2} |f(x)\|^2_{\mathbb{H}_\mathcal{K}} \right] \tag{49}$$

$$\Leftrightarrow \min_{f \in \mathbb{H}_\mathcal{K}} \left[\sum_{i=1}^{N} L(y_i, \sum_{j=1}^{\infty} c_j \phi_j(x_i)) + \frac{\lambda}{2} \sum_{i=1}^{\infty} \frac{c_i^2}{\delta_i} \right] \tag{41}$$

$$\Leftrightarrow \min_{\alpha_0, \alpha} \left[\sum_{i=1}^{N} (1 - y_i(\alpha_0 + \sum_{m=1}^{\infty} c_m \phi_m(x_i)))_+ + \frac{\lambda}{2} \alpha^T \mathcal{K} \alpha \right] \tag{42}$$

where λ to be determined empirically.

Wahba (1990)[1] showed that the solution f is finite-dimensional and has the form

$$f(x) = \alpha_0 + \sum_{i=1}^{N} \alpha_i \mathcal{K}(x, x_i), \tag{50}$$

Note that the RKHS $\mathbb{H}_\mathcal{K}$ provides useful properties such as the reproducing property (20), the evaluation of $f \in \mathbb{H}_\mathcal{K}$ at the point x_i is $f(x_i) =< \mathcal{K}(\cdot, x_i), f >$. Also, due to the property that $< \mathcal{K}(\cdot, x_i), \mathcal{K}(\cdot, x_j) >_{\mathbb{H}_\mathcal{K}} = \mathcal{K}(x_i, x_j)$, the penalty functional can be written as

$$J(f) = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \mathcal{K}(x_i, x_j). \tag{51}$$

3 Methods

3.1 Data dependent kernel

Xiong[5] provided a formulation and algorithm to obtain a data dependent kernel.

3.1.1 Formulation of the kernel

Let $\{(x_i, y_i)\}_{i=1}^N$ be a set of N observations where the input $x_i \in \mathbb{R}^d$ and the output $y_i = \pm 1$, $\forall i = 1, \cdots, N$. It is called the training dataset. Our goal is to fit a kernel SVM classifier to this training dataset. The kernel $\mathcal{K}$ is determined with respect to given data and is called the data dependent kernel.

The data dependent kernel $\mathcal{K}$ is formulated by the conformal transform as

$$\mathcal{K}(x, z) = q(x)\mathcal{K}_0(x, z)q(z) \tag{52}$$

where $x, z \in \mathbb{R}^d$, $\mathcal{K}_0(\cdot, \cdot)$ is called the basic kernel which can be any kernel such as RBF kernel or a polynomial kernel, and $q(\cdot)$ is called the factor function which takes the form of

$$q(x) = \alpha_0 + \sum_{i=1}^{N} \alpha_i \mathcal{K}_1(x, x_i) \tag{53}$$

where $\mathcal{K}_1(x, x_i) = exp(-\gamma_1 \|x - x_i\|^2)$ and α_i's are the combination coefficients.

Denote the kernel matrices corresponding to $\mathcal{K}(\cdot, \cdot)$ and $\mathcal{K}_0(\cdot, \cdot)$ to K and K_0. Then there is a relation between K and K_0 written as

$$K = [q(x_i)K_0(x_i, x_j)q(x_j)]_{N\times N} = QK_0Q \tag{54}$$

where Q is a diagonal matrix with diagonal elements $q(x_i)$, $i = 1, \cdots, N$. Also denote the vectors $q = (q(x_1), q(x_2), \cdots, q(x_N))^T$ and $\alpha = (\alpha(x_1), \alpha(x_2), \cdots, \alpha(x_N))^T$. Then we have $q = K_1\alpha$ where K_1 is an $N \times (N + 1)$ matrix

$$K_1 = \begin{pmatrix} 1 & k_1(x_1, x_1) & \cdots & k_1(x_1, x_N) \\ 1 & k_1(x_2, x_1) & \cdots & k_1(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & k_1(x_N, x_1) & \cdots & k_1(x_N, x_N) \end{pmatrix}. \tag{55}$$

3.1.2 Kernel optimization

Let us fix the basic kernel K_0 and K_1 for the factor function q. The combination coefficients α will be chosen to maximize the class separability of the training data in the mapped feature space which is measured by Fisher scalar. Let us define Fisher scalar

$$F = \frac{tr(S_b)}{tr(S_w)} \tag{56}$$

where S_b represents the "between-class scatter matrix" and S_w "within-class scatter matrix".

Suppose that N_1 training data are labeled as $y_i = 1$ and N_2 training data are labeled as $y_i = -1$, and $N = N_1 + N_2$. Then the basic kernel matrix K_0 can be partitioned as

$$K_0 = \begin{pmatrix} K_{11}^0 & K_{12}^0 \\ K_{21}^0 & K_{22}^0 \end{pmatrix} \tag{57}$$

where the sizes of submatrices K_{11}^0, K_{12}^0, K_{21}^0 and K_{22}^0, are $N_1 \times N_1$, $N_1 \times N_2$, $N_2 \times N_1$, and $N_2 \times N_2$, respectively. Define

$$B_0 = \begin{pmatrix} \frac{1}{N_1}K_{11}^0 & 0 \\ 0 & \frac{1}{N_2}K_{22}^0 \end{pmatrix} - \frac{1}{N}K_0 \tag{58}$$

$$W_0 = diag(k_{11}^0, k_{22}^0, \cdots, k_{NN}^0) - \begin{pmatrix} \frac{1}{N_1}K_{11}^0 & 0 \\ 0 & \frac{1}{N_2}K_{22}^0 \end{pmatrix} \tag{59}$$

and $M_0 = K_1^T B_0 K_1$, $N_0 = K_1^T W_0 K_1$ where the elements of K_1 are aligned in the same order for K_0. Then the Fisher scalar (56) can be written as

$$F(\alpha) = \frac{\alpha^T M_0 \alpha}{\alpha^T N_0 \alpha}. \tag{60}$$

Setting our objective function $F(\alpha)$, the optimization problem

$$\max_{\alpha} F(\alpha) \tag{61}$$

has the solution whenever the matrix N_0 is nonsingular, and the optimal value is λ^* at $\alpha = \alpha^*$ where λ^* is the largest eigenvalue and α^* is the corresponding eigenvector of the system

$$M_0\alpha = \lambda N_0\alpha. \tag{62}$$

Nonsingularity of the matrix N_0 may not be satisfied depending on the problem. If N_0 is noticed to be singular, then modify the problem (62) as

$$M_0\alpha = \lambda(N_0 + \mu I_N)\alpha \tag{63}$$

with the regularization coefficient μ, where I_N is the $N \times N$ identity matrix.

Xiong[5] used gradient ascent method to calculate the optimal α using the below algorithm.

Algorithm 1: Gradient ascent method to calculate α^*

Result: $\alpha^* = \alpha^{maxIterCnt}$

$\{(x_i, y_i)\}_{i=1,\cdots,N}$ where $x_i \in \mathbb{R}^d$, and $y_i \in \{\pm 1\}$;

while $n \leq maxIterCnt$ **do**

  Group the data according to their class labels. Calculate K_0, K_1 first, then M_0, N_0.;

  Initialize $\alpha^0 = (1, 0, \cdots, 0)^T$, and set $n = 0$.;

  Calculate $J_1 = (\alpha^n)^T M_0 \alpha^n$, and $J_1 = (\alpha^n)^T N_0 \alpha^n$.;

  Update α^n by

$$\alpha^{n+1} = \alpha^n + \eta(n)(\frac{1}{J_2}M_0 - \frac{J_1}{J_2^2}N_0)\alpha^n$$

  and then normalize α^{n+1}.;

end

Note) $\eta(t) = \eta_0(1 - \frac{t}{maxIterCnt})$ is decreasing.

3.2 Evaluation of models

	P (predicted)	N (predicted)
P (actual)	True Positive	False Negative
N (actual)	False Positive	True Negative

This table is called *confusion matrix*. Here are the commonly used metrics to evaluate the performance of machine learning models.

$$\text{Accuracy: } \frac{TP + TN}{N} \tag{64}$$

$$\text{Precision: } \frac{TP}{TP + FP} \tag{65}$$

$$\text{Recall: } \frac{TP}{TP + FN} \tag{66}$$

3.3 Multidimensional scaling (MDS)

In order to visualize the performance of a classifier in case of high dimensional inputs in a better way, we consider the projection of the training and test data onto their top two significant dimensions. Here, *multidimensional scaling* (MDS) technique is used.

[ref](https://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html)

4 Application - Tumor/Tumor-free Organs Classification Using Gene Expression Data

4.1 Data set description

GDC portal (https://portal.gdc.cancer.gov/) built up a large open database of cancer genomic/clinical data. A certain cancer project reposits patients' gene expression level data and their labels according to

```
data label: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0.]
```

Unnamed: 0	TP1	TP2	TP3	TP4	TP5	TP6	TP7	TP8	TP9	TP10	...	NT41	NT42	NT43	NT44	NT45	NT46	NT47	NT48	NT49	NT50
ENSG00000000003	1481	3634	578	4994	1592	2623	6175	3788	3617	3536	...	3370	3769	2872	3334	5467	6215	3118	3325	3552	4826
ENSG00000000005	8	58	2	6	16	74	3	332	257	8	...	433	784	2249	9108	1937	474	736	206	782	275
ENSG00000000419	2078	3351	1625	2276	1450	1843	1619	1569	2567	2956	...	1158	1280	1297	1229	1571	2432	1486	1162	1528	1940
ENSG00000000457	1304	1026	1841	3306	2625	1574	2763	1988	1713	957	...	1318	1061	933	980	1038	992	1143	1389	1016	815
ENSG00000000460	711	2661	353	1057	707	1083	777	504	638	1453	...	248	196	231	268	190	232	233	211	216	324
ENSG00000000938	378	295	345	234	406	243	1259	436	448	659	...	714	693	1330	2078	818	442	858	264	1000	1075
ENSG00000000971	4366	1618	2588	1790	1264	1939	4250	5358	9037	1375	...	7369	19277	20395	15823	21997	7797	13500	4588	15627	3963
ENSG00000001036	2820	2914	5077	1638	1807	3991	3326	3076	5213	2579	...	2001	3228	3132	2998	4298	3518	3214	2002	3281	1862
ENSG00000001084	1741	3010	5082	1960	3929	1883	2423	1967	1208	3120	...	2782	2065	4059	4427	2174	2110	3339	1994	3387	2565
ENSG00000001167	1478	3048	2469	3717	3307	3588	2270	2479	1555	3599	...	2025	1514	1661	1880	1445	1628	1533	2558	1596	1593

Figure 1: Input data and labels

the existence of tumors in the certain organ. Figure 1 shows the class labels $y \in \{0, 1\}$, for the convenience and, the data matrix x for the RNA-sequence array (from the TCGA-BRCA project), where its rows and columns corresponds to

- rows: gene labels
- columns: sample indices (TP corresponds to the tumor(-primary) sample with data label 1, and NT corresponds to the non-tumor sample with data label 0).

Among the 50 TP samples and 50 NT samples, training and test sets are divided with the ratio of 6:4. 10 most relevant genes(rows) are chosen by the BW score on gene j calculated as

$$g(j) = \frac{\sum_{k=1}^{2} m_k(\bar{x}_k(j) - \bar{x})^2}{\sum_{k=1}^{2} \sum_{i \in C_k} (x_i(j) - \bar{x}_k)^2} \tag{67}$$

where C_k denotes the index set of the k-th class ($k = 1, 2$), m_k is the number of samples in C_k, and $\bar{x}_k(j)$ and $\bar{x}$ represent the average value within the k-th class and of the entire training sample on gene j, respectively. High value in g implies that the gene j better separates the different classes and gathers within a class.

4.2 Software

Python v.3.9 platform and its libraries

- scikit-learn : generate inputs, SVM training (https://scikit-learn.org/stable/modules/svm.html), multidimensional scaling (https://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html)
- matplotlib.pyplot: visualize the data and performances
- pandas: retrive R dataframe into python

R and its library

- tcgaBiolinks : retrieve tcga data from GDC portal. (https://portal.gdc.cancer.gov/)

4.3 Results

Three different kernels (linear, gaussian, data-dependent) are individually used for training SVMs, measured by performance metrics in sec 3.2 and plotted in $\mathbb{R}^2$ by projecting data using the MDS technique.

The setups for the data-dependent kernel $K(x,y) = q(x)^T K_0(x,y)q(y)$ where $q(x) = \sum_{i=1}^{M} \alpha_0 + \sum_{i=1}^{M} \alpha_i K_1(x, x_i)$. It is optimized using the algorithm of Xiong, and the hyper-parameters for the optimization are as below:

- K_0, K_1 are gaussian kernels with the fixed parameter $\gamma = \frac{1}{M}$
- $\eta(i) = 1 * (1 - \frac{i}{maxIterCnt})$ for the step size of each gradient-ascent iteration
- $maxIterCnt = 1000$ but terminated when the objective function starts to decrease at the 894th step with the objective function value 0.01975309.

It is different from the observation from Xiong [6] that the Within-Class error is typically almost zero. Even, the objective function value which is given as a ratio of the Between-Class error and the Within-Class error is still less than 1, which foretells the performance of the kernel would be bad.

The performance of individual kernels on 18 test samples which are labeled as 1 and 22 test samples which are labeled as 0 are given in Table 4.3. Unfortunately, the linear kernel could only discriminate the test data but it was still not perfect. It fails at classifying some samples of the label 1. Whereas, the other two non-linear kernels based on the gaussian kernel didn't work at all. The values for precision and recall show that the classifiers output 0 for the entire data.

	Linear SVM	Rbf kernel SVM	Data-driven SVM
Accuracy (%)	80.00	45.00	45.00
Precision	0.92	0.45	0.45
Recall	0.61	1.00	1.00

Table 1: Performances of three different kernels

The visualization of high-dimensional input is aided by the MDS technique which projects the test data into $\mathbb{R}^2$. Test data of two different classes and their support vectors are shown in Figure 2. Blue dots represent the support vectors, while red and yellow dots are label 0 and 1, respectively. Visualization gives much easier explanation that the gaussian kernel selects a wrong support vector on the upper right side of the second figure. The sklearn svm fitting function could not find support vectors. The possible reason is due to the mistakes on providing test data in case of user-prescribed kernel.

References

[1] *7. Finite-Dimensional Approximating Subspaces*, pages 95–99. doi:10.1137/1.9781611970128.ch7. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611970128.ch7.

[2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Support Vector Machines and Flexible Discriminants*, pages 417–458. Springer New York, New York, NY, 2009. isbn:978-0-387-84858-7. doi:10.1007/978-0-387-84858-7_12. URL https://doi.org/10.1007/978-0-387-84858-7_12.

[3] Bernhard Schölkopf and Alexander J. Smola. *Designing Kernels*, pages 407–426. 2001.

[4] Martin J. Wainwright. *Reproducing kernel Hilbert spaces*, page 383–415. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019. doi:10.1017/9781108627771.012.

[5] Huilin Xiong, M.N.S. Swamy, and M.O. Ahmad. Optimizing the kernel in the empirical feature space. *IEEE Transactions on Neural Networks*, 16(2):460–474, 2005. doi:10.1109/TNN.2004.841784.

[6] Huilin Xiong, Ya Zhang, and Xue-Wen Chen. Data-dependent kernel machines for microarray data classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(4):583–595, 2007. doi:10.1109/tcbb.2007.1048.

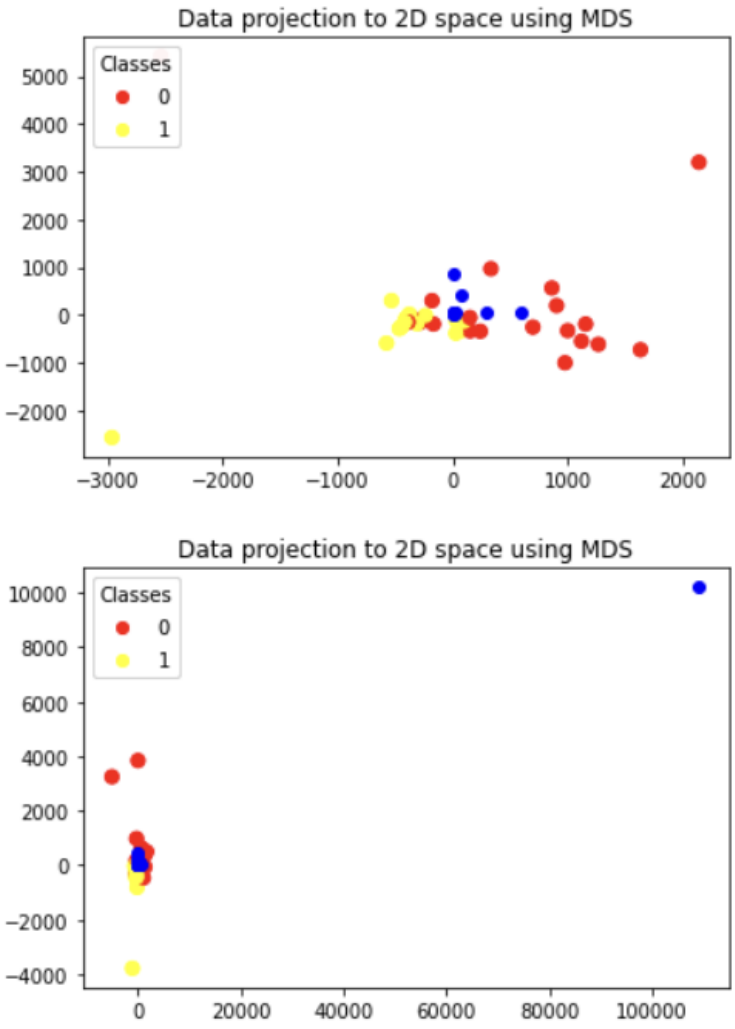


Figure 2: Projection of data onto 2 significant input features, and support vectors