Machine Learning (ML)

Prerequisites

Definition (**Real Vector Space**). *A set $\mathcal{H}$ is called a vector space over $\mathbb{R}$ if addition and scalar multiplication are defined, and satisfy $\forall \mathbf{x}, \mathbf{x}', \mathbf{x}'' \in \mathcal{H}$ and $\lambda, \lambda' \in \mathbb{R}$:*

$$\begin{aligned}
&\mathbf{x} + (\mathbf{x}' + \mathbf{x}'') = (\mathbf{x} + \mathbf{x}') + \mathbf{x}'',\\
&\mathbf{x} + \mathbf{x}' = \mathbf{x}' + \mathbf{x} \in \mathcal{H},\\
&0 \in \mathcal{H}, \mathbf{x} + 0 = \mathbf{x},\\
&-\mathbf{x} \in \mathcal{H}, \mathbf{x} - \mathbf{x} = 0,\\
&\lambda \mathbf{x} \in \mathcal{H},\\
&1\mathbf{x} \in \mathcal{H},\\
&\lambda\left(\lambda' \mathbf{x}\right) = \left(\lambda\lambda'\right)\mathbf{x},\\
&\lambda\left(\mathbf{x} + \mathbf{x}'\right) = \lambda\mathbf{x} + \lambda\mathbf{x}'
\end{aligned}$$

Definition (**Norm**). *A function $\|\cdot\| : \mathcal{H} \to \mathbb{R}_0^+$ that for all $\mathbf{x}, \mathbf{x}' \in \mathcal{H}$ and $\lambda \in \mathbb{R}$ satisfies:*

$$\begin{aligned}
&\|\mathbf{x} + \mathbf{x}'\| \leq \|\mathbf{x}\| + \|\mathbf{x}'\|,\\
&\|\lambda\mathbf{x}\| = |\lambda|\|\mathbf{x}\|,\\
&\|\mathbf{x}\| > 0 \textit{ if } \mathbf{x} \neq 0,
\end{aligned}$$

is called a norm on $\mathcal{H}$.

Definition (**Dot Product**). *A dot product on a vector space $\mathcal{H}$ is a symmetric bilinear form,*

$$\begin{aligned}
\langle .,. \rangle : \mathcal{H} \times \mathcal{H} &\to \mathbb{R}\\
(\mathbf{x}, \mathbf{x}') &\mapsto \langle \mathbf{x}, \mathbf{x}' \rangle
\end{aligned}$$

that is strictly positive definite.

Definition (**Normed Space and Dot Product Space**). *A normed space is a vector space endowed with a norm; a dot product space (pre-Hilbert space) is a vector space endowed with a dot product.*

Definition (**Cauchy Sequence**). *A sequence $(\mathbf{x}_i)_i := (\mathbf{x}_i)_{i\in\mathbb{N}} = (\mathbf{x}_1, \mathbf{x}_2, \dots)$ in a normed space $\mathcal{H}$ is said to be a Cauchy sequence if for every $\epsilon > 0$, there exists an $n \in \mathbb{N}$ such that for all $n', n'' > n$, $\|\mathbf{x}_{n'} - \mathbf{x}_{n''}\| < \epsilon$.*

Definition (**Hilbert Space**). *A space $\mathcal{H}$ is called complete if all Cauchy sequences in the space converge. A Hilbert space is a complete dot product space. Hilbert spaces have infinite dimensionality.*

Example (**Hilbert Space of Functions**). *Let $C[a,b]$ denote the real-valued continuous functions on the interval $[a,b]$ For $f, g \in C[a,b]$,*

$$\langle f, g \rangle := \int_a^b f(x)g(x)dx$$

defines a dot product. The completion of $C[a,b]$ in the corresponding norm is the Hilbert space $L_2[a,b]$ of measurable functions that are square integrable:

$$\int_a^b f^2(x)dx < \infty$$

Elements of Statistical Learning Theory

Learning problem

In two-class pattern recognition, we seek to infer a function:

$$f : \mathcal{X} \to \{\pm 1\}$$

Statistical learning theory makes the assumption that the data are generated by sampling from an unknown underlying distribtuion $P(x,y)$. The learning problem then consists in minimizing the *risk*:

$$R[f] = \int_{\mathcal{X}\times\mathcal{Y}} \underbrace{c(x, y, f(x))}_{\text{loss function}} dP(x,y)$$

We do not know P. We do know the training data, which are sample from P. This leads to the empirical risk:

$$R_{emp}[f] = \frac{1}{m}\sum_{i=1}^{m} c(x_i, y_i, f(x_i))$$

For the purpose of bounding the probability:

$$P\left\{\sup_{f\in\mathcal{F}}\left(R[f] - R_{emp}[f]\right) > \epsilon\right\}$$

The function class $\mathcal{F}$ is effectively finite. Let $Z_{2m} := \left((x_1, y_1), \dots, (x_{2m}, y_{2m})\right)$ be the given $2m$-sample. Denote by $\mathcal{N}(\mathcal{F}, Z_{2m})$ the cardinality of $\mathcal{F}$ when restricted to $\{x_1, \dots, x_{2m}\}$, that is, the number of functions from $\mathcal{F}$ that can be distinguished from their values on $\{x_1, \dots, x_{2m}\}$. Denote the maximum number of functions that can be distinguished as $\mathcal{N}(\mathcal{F}, 2m)$. The function $\mathcal{N}(\mathcal{F}, m)$ is referred to as the *shattering coefficient*. It measures the number of ways that the function class can separate the patterns into two classes.

VC Dimension and Other Capacity Concepts

By taking a supremum over all possible samples:

$$G_{\mathcal{F}}(m) = \max_{(x_1,y_1),\dots,(x_m,y_m)\in\mathcal{X}\times\{\pm\}} \ln \mathcal{N}\left(\mathcal{F}, (x_1, y_1), \dots, (x_m, y_m)\right)$$

this leads to the *growth function*. If $\mathcal{F}$ is as rich as possible, so that for any sample of size m, they can be separated in all 2^m possible ways (i.e., they can be shattered), then:

$$G_{\mathcal{F}}(m) = m \cdot \ln(2)$$

There exists some *maximal* m for which it is satisfied (*VC dimension*). The VC dimension can be shown to be $N+1$ for hyperplanes in $\mathbb{R}^N$.

Linear regression

Method for predicting a real-valued output (also called the **dependent variable** or **target**) $y \in \mathbb{R}$ given a vector of real-valued inputs (also called **independent variables, explanatory variables** or **covariates**) $\mathbf{x} \in \mathbb{R}^D$.

Linear regression usually refers to a model of the form:

$$p\left(y \mid \mathbf{x}, \theta\right) = \mathcal{N}\left(y \mid \mathbf{w}^T\mathbf{x} + b, \sigma^2\right)$$

where $\theta = \left(b, \mathbf{w}, \sigma^2\right)$ are the parameters. The **residual sum of squares** is given by:

$$\frac{1}{2}\sum_{n=1}^{N}\left(y_n - \mathbf{w}^T\mathbf{x}_n\right)^2 = \frac{1}{2}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 = \frac{1}{2}\left(\mathbf{X}\mathbf{w} - \mathbf{y}\right)^T\left(\mathbf{X}\mathbf{w} - \mathbf{y}\right)$$

Setting the gradient to zero and solving gives:

$$\begin{aligned}
&\nabla_{\mathbf{w}}\left(\frac{1}{2}\left(\mathbf{X}\mathbf{w} - \mathbf{y}\right)^T\left(\mathbf{X}\mathbf{w} - \mathbf{y}\right)\right) =\\
&\nabla_{\mathbf{w}}\left(\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - \mathbf{w}^T\mathbf{X}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\mathbf{w} + \mathbf{y}^T\mathbf{y}\right) =\\
&\nabla_{\mathbf{w}}\left(\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{w}^T\mathbf{X}^T\mathbf{y} + \mathbf{y}^T\mathbf{y}\right) =\\
&2\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{X}^T\mathbf{y} = 0\\
&\boxed{\hat{\mathbf{w}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}}
\end{aligned}$$

Ridge regression/l_2 regularization/Tikhonov regularization

Maximum likelihood estimation can result in overfitting. The main solution to overfitting is to use **regularization**, which means to add a penalty term to the empirical risk. Thus we optimize:

$$\boxed{\hat{\mathbf{w}}_{map} = \operatorname{argmin} \frac{1}{2}\left(\mathbf{X}\mathbf{w} - \mathbf{y}\right)^T\left(\mathbf{X}\mathbf{w} - \mathbf{y}\right) + \lambda\|\mathbf{w}\|^2}$$

$$\boxed{\hat{\mathbf{w}}_{map} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y}}$$

Feature extraction

In general, a straight line will not provide a good fit. We can always apply a nonlinear transformation to the input features by replacing $\mathbf{x}$ with $\phi(\mathbf{x})$. For example, we can use a polynomial transform, which in 1D is given by $\phi(x) = \left[1, x, x^2, x^3, \dots\right]$. The model becomes:

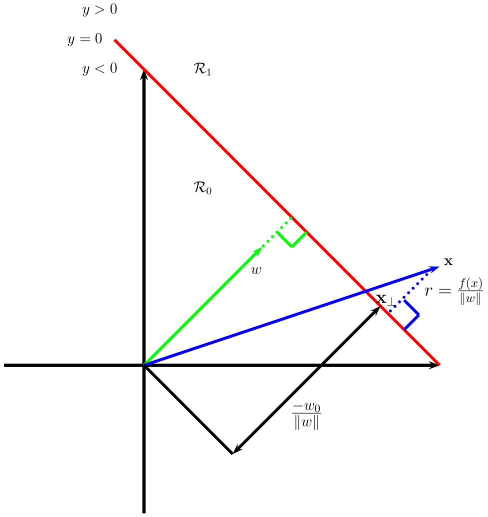
$$\boxed{f(\mathbf{x}; \theta) = \mathbf{W}\phi(\mathbf{x}) + \mathbf{b}}$$

Support vector machines (SVMs)

Consider a **binary classifier** of the form $h(\mathbf{x}) = \text{sign}\left(f(\mathbf{x})\right)$ where the decision boundary is given by:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$$

Labels are -1 and +1 rather than 0 and 1.



The distance of a point to the decision boundary is:

$$\mathbf{x} = \mathbf{x}_\perp + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$$

where r is the distance of $\mathbf{x}$ from the decision boundary whose normal vector is $\mathbf{w}$, and $\mathbf{x}_\perp$ is the orthogonal projection of $\mathbf{x}$ onto this boundary.

Definition. ***(Geometrical Margin)*** *For a hyperplane* $\{\mathbf{x} \in \mathcal{H} \mid \langle \mathbf{w}, \mathbf{x}\rangle + b = 0\}$, *we call*

$$\rho_{(\mathbf{w},b)}(\mathbf{x}, y) := y\left(\langle \mathbf{w}, \mathbf{x}\rangle + b\right)/\|\mathbf{w}\|$$

the geometrical margin of the point $(\mathbf{x}, y) \in \mathcal{H} \times \{\pm 1\}$. *The minimum value*

$$\rho_{(\mathbf{w},b)} := \min_{i=1,\dots,m} \rho_{(\mathbf{w},b)}(\mathbf{x}_i, y_i)$$

shall be called the geometrical margin of $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$.

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{w}^T\mathbf{x} + b \\ &= \mathbf{w}^T\mathbf{x}_\perp + b + r\frac{\mathbf{w}^T\mathbf{w}}{\|\mathbf{w}\|} \\ &= \mathbf{w}^T\mathbf{x}_\perp + b + r\|\mathbf{w}\| \end{aligned}$$

Since $0 = f(\mathbf{x}_\perp) = \mathbf{w}^T\mathbf{x}_\perp + b$, we have $f(\mathbf{x}) = r\|\mathbf{w}\|$ and hence $r = \frac{f(\mathbf{x})}{\|\mathbf{w}\|}$. We also require $f(\mathbf{x}_n)y_n > 0$ (ensure each point is on the correct side of the boundary). We want:

$$\underbrace{\max_{\mathbf{w},b} \frac{1}{\|\mathbf{w}\|}}_{\text{maximize the distance}} \quad \underbrace{\min_{n=1}^{N} \left[y_n \left(\mathbf{w}^T\mathbf{x}_n + b \right) \right]}_{\text{of the closest point}}$$

Let us define the scale factor such that $y_n f_n = 1$ for the point that is closest to the decision boundary. Hence we require $y_n f_n \geq 1$ for all n. Maximizing $1/\|\mathbf{w}\|$ is equivalent to minimizing $\|\mathbf{w}\|^2$. Thus:

$$\boxed{\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{s.t. } y_n\left(\mathbf{w}^T\mathbf{x}_n + b\right) \geq 1}$$

$N + D + 1$ variables subject to N constraints (**primal problem**).

In convex optimization, for every primal problem we can derive a **dual problem**. Let $\alpha \in \mathbb{R}^N$ be the dual variables corresponding to Lagrange multipliers that enforce N inequality constraints:

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{n=1}^{N} \alpha_n \left(y_n \left(\mathbf{w}^T\mathbf{x}_n + b \right) - 1 \right)$$

We have:

$$\begin{aligned} \nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}, b, \alpha) &= \mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n \\ \frac{\partial}{\partial b}\mathcal{L}(\mathbf{w}, b, \alpha) &= -\sum_{n=1}^{N} \alpha_n y_n \end{aligned}$$

and hence:

$$\begin{aligned} \hat{\mathbf{w}} &= \sum_{n=1}^{N} \hat{\alpha}_n y_n \mathbf{x}_n \\ 0 &= \sum_{n=1}^{N} \hat{\alpha}_n y_n \end{aligned}$$

Plugging these into the Lagrangian:

$$\begin{aligned} \mathcal{L}\left(\hat{\mathbf{w}}, \hat{b}, \alpha\right) &= \frac{1}{2}\hat{\mathbf{w}}^T\hat{\mathbf{w}} - \sum_{n=1}^{N} \alpha_n y_n \hat{\mathbf{w}}^T\mathbf{x}_n - \sum_{n=1}^{N} \alpha_n y_n \hat{b} + \sum_{n=1}^{N} \alpha_n \\ &= \frac{1}{2}\hat{\mathbf{w}}^T\hat{\mathbf{w}} - \hat{\mathbf{w}}^T\hat{\mathbf{w}} - 0 + \sum_{n=1}^{N} \alpha_n \\ &= \boxed{-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j + \sum_{n=1}^{N} \alpha_n} \end{aligned}$$

We want to maximize this wrt α subject to the constraints $\sum_{n=1}^{N} \alpha_n y_n = 0$ and $\alpha_n \geq 0$ (N variables).
KKT conditions:

$$\begin{aligned} \alpha_n &\geq 0 \\ y_n f(\mathbf{x}) - 1 &\geq 0 \\ \alpha_n \left(y_n f(\mathbf{x}) - 1 \right) &\geq 0 \end{aligned}$$

Hence either $\alpha_n = 0$ or $y_n\left(\hat{\mathbf{w}}^T\mathbf{x}_n + \hat{b}\right) = 1$ is active (sample n lies on the decision boundary; **support vector**).
Denote by $\mathcal{S}$ the set of support vectors. To perform prediction:

$$\boxed{f(\mathbf{x}; \hat{\mathbf{w}}, \hat{b}) = \hat{\mathbf{w}}^T\mathbf{x} + b = \sum_{n\in\mathcal{S}} \alpha_n y_n \mathbf{x}_n^T\mathbf{x} + b}$$

Soft margin classifiers

If data is not linearly separable, there will be no feasible solution in which $y_n f_n \geq 1$ for all n. We therefore introduce **slack variables** $\xi_n \geq 0$ and replace the hard constraints $y_n f_n \geq 0$ with $y_n f_n \geq 1 - \xi_n$. The new objective becomes:

$$\boxed{\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \quad \text{s.t. } \xi_n \geq 0, y_n\left(\mathbf{w}^T\mathbf{x}_n + b\right) \geq 1 - \xi_n}$$

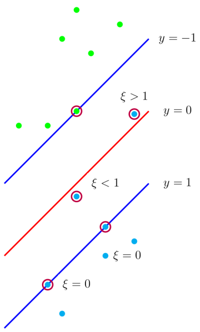
C is a hyperparameter controlling how many points violate the margin constraints (if $C = \infty$ we recover the unregularized, hard-margin classifier).
The corresponding Lagrangian for the soft margin classifier becomes:

$$\begin{aligned} \mathcal{L}(\mathbf{w}, b, \alpha, \xi, \mu) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n - \\ \sum_{n=1}^{N}\alpha_n\left(y_n\left(\mathbf{w}^T\mathbf{x}_n + b\right) - 1 + \xi_n\right) - \sum_{n=1}^{N}\mu_n\xi_n \end{aligned}$$

where $\alpha_n \geq 0$ and $\mu_n \geq 0$ are the Lagrange multipliers.
Optimizing out $\mathbf{w}$, b and ξ gives the dual form identical to the hard margin case. However, the KKT conditions imply:

$$\begin{aligned} 0 \leq \alpha_n \leq C \\ \sum_{n=1}^{N} \alpha_n y_n = 0 \end{aligned}$$

If $\alpha_n = 0$, the point is ignored.
If $0 < \alpha_n < C$, then $\xi_n = 0$, so the point lies on the margin.
If $\alpha_n = C$, the point can either be correctly classified if $\xi_n \leq 1$, or misclassified if $\xi_n > 1$. Hence $\sum_n \xi_n$ is an upper bound on the number of misclassified points.



Kernels

Kernel trick

The principal benefit of the dual problem is that we can replace all inner product operations $\mathbf{x}^T\mathbf{x}'$ with a call to a positive definite kernel function $\mathcal{K}(\mathbf{x},\mathbf{x}')$. The kernel trick allows us to avoid having to deal with an explicit feature representation of our data. In particular:

$$f(\mathbf{x}) = \hat{\mathbf{w}}^T\mathbf{x} + b = \sum_{n\in\mathcal{S}} \alpha_n y_n \mathbf{x}_n^T\mathbf{x} + \hat{b} = \sum_{n\in\mathcal{S}} \alpha_n y_n \mathcal{K}(\mathbf{x}_n,\mathbf{x}') + \hat{b}$$

A simple type of similarity measure is a *dot product*. For instance, given two vectors $\mathbf{x},\mathbf{x}' \in \mathbb{R}^N$, the canonical dot product is defined as:

$$\langle \mathbf{x},\mathbf{x}'\rangle := \sum_{i=1}^{N} [\mathbf{x}]_i [\mathbf{x}']_i$$

Patterns could be any kind of object. In order to be able to use a dot product as a similarity measure, we therefore first need to represent the patterns as vectors in some dot product space $\mathcal{H}$:

$$\begin{aligned} \mathbf{\Phi} &: \mathcal{X} \to \mathcal{H} \\ x &\mapsto \mathbf{x} := \mathbf{\Phi}(x) \end{aligned}$$

The space $\mathcal{H}$ is called a *feature space*. A similarity measure from the dot product in $\mathcal{H}$:

$$k(x,x') := \langle \mathbf{x},\mathbf{x}'\rangle = \langle \mathbf{\Phi}(x),\mathbf{\Phi}(x')\rangle$$

In binary classification, two labels (outputs) can either be identical or different. Let us consider a similarity measure of the form:

$$\begin{aligned} k &: \mathcal{X}\times\mathcal{X} \to \mathbb{R} \\ (x,x') &\mapsto k(x,x') \end{aligned}$$

that is, a function that given two patterns returns a real number characterizing their similarity.

Product Features

For $\mathcal{X}$ a subset of the vector space $\mathbb{R}^N$, $N = d = 2$, dot products in $\mathcal{H}$ for the map:

$$\mathbf{\Phi} : ([x]_1, [x]_2) \mapsto \left([x]_1^2, [x]_2^2, [x]_1[x]_2, [x]_2[x]_1\right)$$

take the form:

$$\langle \mathbf{\Phi}(x),\mathbf{\Phi}(x')\rangle = [x]_1^2[x']_1^2 + [x]_2^2[x']_2^2 + 2[x]_1[x]_2[x']_1[x']_2 = \langle x,x'\rangle^2$$

In other words, the desired kernel is simply the square of the dot product in input space. The same works for arbitrary $N, d \in \mathbb{N}$.

Positive Definite Kernels

The results in this section hold for data drawn from domains which need no structure.

Definition. ***(Gram Matrix)*** *Given a function $k : \mathcal{X}^2 \to \mathbb{R}$ and patterns $x_1,\ldots,x_m \in \mathcal{X}$, the $m\times m$ matrix K with elements $K_{ij} := k(x_i,x_j)$ is called the Gram matrix of k.*

Definition. ***(Positive Definite Matrix)*** *A real symmetric[1] $m\times m$ matrix K satisfying $\sum_{i,j} c_i c_j K_{ij} \geq 0$ for all $c_i \in \mathbb{R}$ is called positive definite.*

Definition. ***(Positive Definite Kernel)*** *A function k on $\mathcal{X}\times\mathcal{X}$ gives rise to positive definite Gram matrix is called a positive definite kernel.*

The Reproducing Kernel Map

We describe the construction of a dot product on the function space such that $k(x,x') = \langle \mathbf{\Phi}(x),\mathbf{\Phi}(x')\rangle$. Here $\mathbf{\Phi}(x)$ denotes the function that assigns the value $k(x',x)$ to $x' \in \mathcal{X}$, i.e., $\mathbf{\Phi}(x)(\cdot) = k(\cdot,x)$.
We define a vector space by taking linear combinations of the form:

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot,x_i)$$

Here, $m \in \mathbb{N}$, $\alpha_i \in \mathbb{R}$ and $x_1,\ldots,x_m \in \mathcal{X}$ are arbitrary.
Next, we define a dot product between f and another function:

$$g(\cdot) = \sum_{j=1}^{m'} \beta_j k\left(\cdot,x_j'\right)$$

as:

$$\langle f,g\rangle = \sum_{i=1}^{m}\sum_{j=1}^{m'} \alpha_i\beta_j k\left(x_i,x_j'\right) \overset{\text{sym.}}{=} \langle g,f\rangle$$

Moreover,

$$\langle f,f\rangle = \sum_{i,j=1}^{m} \alpha_i\alpha_j k\left(x_i,x_j\right) \overset{\text{pd}}{\geq} 0$$

In proving that it qualifies as a dot product:

$$\langle k(\cdot,x), f\rangle = f(x)$$

In particular:

$$\langle k(\cdot,x), k(\cdot,x')\rangle = k(x,x')$$

By virtue of these properties, pd kernels are also called *reproducing kernels*. The above reasoning has shown that any positive definite kernel can be thought as a dot product in another space. In view of:

$$\begin{aligned} k &: \mathcal{X}\times\mathcal{X} \to \mathbb{R} \\ (x,x') &\mapsto k(x,x') \end{aligned}$$

the reproducing kernel property:

$$\langle k(\cdot,x), k(\cdot,x')\rangle = k(x,x')$$

amounts to:

$$\langle \mathbf{\Phi}(x),\mathbf{\Phi}(x')\rangle = k(x,x')$$

Reproducing Kernel Hilbert Spaces

Definition (Reproducing Kernel Hilbert Space). *$\mathcal{H}$ is called a reproducing kernel Hilbert space endowed with the dot product and the norm ($\|f\| := \sqrt{\langle f,f\rangle}$) if there exists a function $k : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ with the following properties:*

1. *(Reproducing Property) k has the reproducing property*
$$\langle f, k(x,\cdot)\rangle = f(x) \quad \text{for all } f \in \mathcal{H}$$
2. *(Closed Space) k spans $\mathcal{H}$, i.e., $\mathcal{H} = \operatorname{span}\{k(x,\cdot) \mid x \in \mathcal{X}\}$ (completion of set).*

RKHS uniquely determines k.

The Representer Theorem

The significance of the Representer Theorem is that although we might be trying to solve an optimization problem in an infinite-dimensional space $\mathcal{H}$, it states that the solution lies in the span of m particular kernels.

Theorem (Representer Theorem). *Each minimizer $f \in \mathcal{H}$ of the regularized risk:*

$$\underbrace{c\left((x_1,y_1,f(x_1)),\ldots,(x_m,y_m,f(x_m))\right)}_{\text{arbitrary loss function}} + \underbrace{\Omega\left(\|f\|_{\mathcal{H}}\right)}_{\text{strictly monotonic increasing funcion}}$$

admits a representation of the form:

$$f(x) = \sum_{i=1}^{m} \alpha_i k(x_i,x)$$

The Empirical Kernel Map

It is possible to approximate the map $\mathbf{\Phi}$ by only evaluating it on any give set of points.

Definition (Empirical Kernel Map). *For a given set $\{z_1,\ldots,z_n\} \subset \mathcal{X}$, $n \in \mathbb{N}$, we call*

$\mathbf{\Phi}_n : \mathbb{R}^N \to \mathbb{R}^n$ where $x \mapsto k(\cdot,x)|_{\{z_1,\ldots,z_n\}} = (k(z_1,x),\ldots,k(z_n,x))^T$ the empirical kernel map. Consider $\{z_1,\ldots,z_n\} = \{x_1,\ldots,x_m\}$. To turn $\mathbf{\Phi}_m$ into a feature map, we need to endow $\mathbb{R}^m$ with a dot product such that:

$$k(x,x') = \langle \mathbf{\Phi}_m(x),\mathbf{\Phi}_m(x')\rangle$$

We use $\langle\cdot,\cdot\rangle = \langle\cdot, M\cdot\rangle$ with M being a pd matrix.

[1] $k(x_i,x_j) = k(x_j,x_i)$

Support Vector Regression

An analog of the soft margin is constructed in the space of the target values y by using Vapnik's *ϵ-insensitive loss function*. This quantifies the loss incurred by predicting $f(\mathbf{x})$ instead of y:

$$c(x, y, f(x)) := |y - f(\mathbf{x})|_\epsilon := \max\left\{0, |y - f(\mathbf{x})| - \epsilon\right\}$$

Any point lying inside an ϵ-tube around the prediction is not penalized.
To estimate a linear regression:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

one minimizes:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{m} |y_i - f(\mathbf{x}_i)|_\epsilon$$

We transform this into a constrained optimization problem by introducing two types of slack variables for the two cases $f(\mathbf{x}_i) - y_i > \epsilon$ and $y_i - f(\mathbf{x}_i) > \epsilon$. We denote them by ξ and ξ^*, respectively, and collectively, $\xi^{(*)}$.

$$\min_{\mathbf{w}, \xi^{(*)}, b} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \left(\xi_i + \xi_i^*\right)$$

$$\begin{aligned} \text{s.t. } & f(\mathbf{x}_i) - y_i \le \epsilon + \xi_i, \\ & y_i - f(\mathbf{x}_i) \le \epsilon + \xi_i^*, \\ & \underbrace{\xi_i, \xi_i^*}_{\xi_i^{(*)}} \ge 0 \end{aligned}$$

Standard quadratic program in $2N + D + 1$ variables.
By forming the Lagrangian from the objective function and the corresponding constraints, by introducing a dual set of variables,

$$\begin{aligned} \mathcal{L}\left(\mathbf{w}, \xi^{(*)}, \alpha, \eta\right) = & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C \sum_{i=1}^{m} \left(\xi_i + \xi_i^*\right) - \sum_{i=1}^{m} \left(\eta_i \xi_i + \eta_i^* \xi_i^*\right) \\ & - \sum_{i=1}^{m} \alpha_i \left(\epsilon + \xi_i + y_i \underbrace{- \langle \mathbf{w}, \mathbf{x}_i \rangle - b}_{-f(\mathbf{x}_i)}\right) \\ & - \sum_{i=1}^{m} \alpha_i^* \left(\epsilon + \xi_i^* - y_i + \underbrace{\langle \mathbf{w}, \mathbf{x}_i \rangle + b}_{f(\mathbf{x}_i)}\right) \end{aligned}$$

where $\alpha_i^{(*)}, \eta_i^{(*)} \ge 0$ are the dual variables (or Lagrange multipliers).

$$\partial_b \mathcal{L} = \sum_{i=1}^{m} \left(\alpha_i - \alpha_i^*\right) = 0$$

$$\nabla \mathbf{w} \mathcal{L} = \mathbf{w} - \sum_{i=1}^{m} \left(\alpha_i^* - \alpha_i\right) \mathbf{x_i} = 0$$

$$\partial_{\xi_i^{(*)}} \mathcal{L} = C - \alpha_i^{(*)} - \eta_i^{(*)} = 0$$

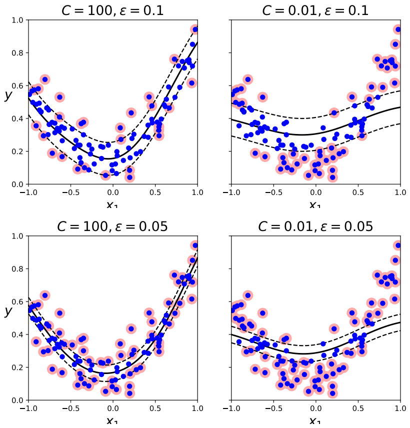
Substituting, the dual optimization problem:

$$\begin{aligned} \max_{\alpha^{(*)}} & - \frac{1}{2} \sum_{i,j=1}^{m} \left(\alpha_i - \alpha_i^*\right)\left(\alpha_j - \alpha_j^*\right) \{\mathbf{x}_i, \mathbf{x}_j\} \\ & - \epsilon \sum_{i=1}^{m} \left(\alpha_i^* + \alpha_i^*\right) + \sum_{i=1}^{m} y_i \left(\alpha_i^* - \alpha_i^*\right) \\ \text{subject to } & \sum_{i=1}^{m} \left(\alpha_i^* + \alpha_i^*\right) = 0, \alpha_i, \alpha_i^* \in [0, C] \end{aligned}$$

Thus:

$$\boxed{f(\mathbf{x}) = \sum_{i=1}^{m} (\alpha_i^* - \alpha_i)\langle \mathbf{x}_i, \mathbf{x} \rangle + b}$$

The vector α is sparse, meaning that may of its entries are equal to 0. This is because the loss doesn't care about error which are small than ϵ. The degree of sparsity is controlled by C and ϵ.



Kernel PCA

Standard PCA

Given a set of observations $x_i \in \mathbb{R}^N$, $i = 1, \ldots, m$, which are centered, $\sum_{i=1}^m x_i = 0$, PCA finds the principal axes by diagonalizing the covariance matrix:

$$C = \frac{1}{m} \sum_{j=1}^{m} x_j x_j^T$$

C is positive definite and can thus be diagonalized with nonnegative eigenvalues:

$$\lambda v = Cv = \frac{1}{m} \sum_{j=1}^{m} \langle x_j, v \rangle x_j$$

All solutions v lie in the span of $x_1, \ldots, x_m$, hence:

$$\lambda \langle x_i, v \rangle = \langle x_i, Cv \rangle$$

for all $i = 1, \ldots, m$.

Kernel PCA

We are not interested in principal components in input space but of variables or features, which are nonlinearly related to the input variables.

$$\begin{aligned} \mathbf{\Phi} : \mathcal{X} & \to \mathcal{H} \\ x & \mapsto \mathbf{x} := \mathbf{\Phi}(x) \end{aligned}$$

Again we are dealing with centered data. The covariance matrix takes the form:

$$C = \frac{1}{m} \sum_{j=1}^{m} \mathbf{\Phi}(x_j)\mathbf{\Phi}(x_j)^T$$

All solutions $\mathbf{v}$ ($\lambda \mathbf{v} = \mathbf{C}\mathbf{v}$) lie in the span of $\mathbf{\Phi}(x_1), \ldots, \mathbf{\Phi}(x_m)$. We may consider the set of equations:

$$\lambda \langle \mathbf{\Phi}(x_n), \mathbf{v} \rangle = \langle \mathbf{\Phi}(x_n), \mathbf{C}\mathbf{v} \rangle$$

and there exists coefficients α_i such that:

$$\mathbf{v} = \sum_{i=1}^{m} \alpha_i \mathbf{\Phi}(x_i)$$

Combining:

$$m\lambda K\alpha = K^2\alpha$$

We solve the dual eigenvalue problem:

$$\boxed{m\lambda\alpha = K\alpha}$$

Let $\lambda_1 \ge \cdots \ge \lambda_m$ denote the eigenvalues of K, and $\alpha^1, \ldots, \alpha^m$ the corresponding complete set of eigenvectors. Let x be a test point, with an image $\mathbf{\Phi}(x)$. Then:

$$\langle \mathbf{v}^n, \mathbf{\Phi}(x) \rangle = \sum_{i=1}^{m} \alpha_i^n \langle \mathbf{\Phi}(x_i), \mathbf{\Phi}(x) \rangle$$

There is a way to compute the mean of the mapped observations in $\mathcal{H}$:

$$\tilde{K}_{ij} = (K - 1_m K - K 1_m + 1_m K 1_m)_{ij}$$

using $(1_m)_{ij} := 1/m$ for all i, j.