Machine Learning and Neural Networks (MATH3431) Epiphany term, 2024

Handout 8: Kernel methods

Lecturer & author: Georgios P. Karagiannis georgios.karagiannis@durham.ac.uk

Aim. To introduce the ideas of learning machines by introducing data into high-dimensional feature spaces for accuracy gains; introduce the kernel trick, and kernel functions.

Reading list & references:

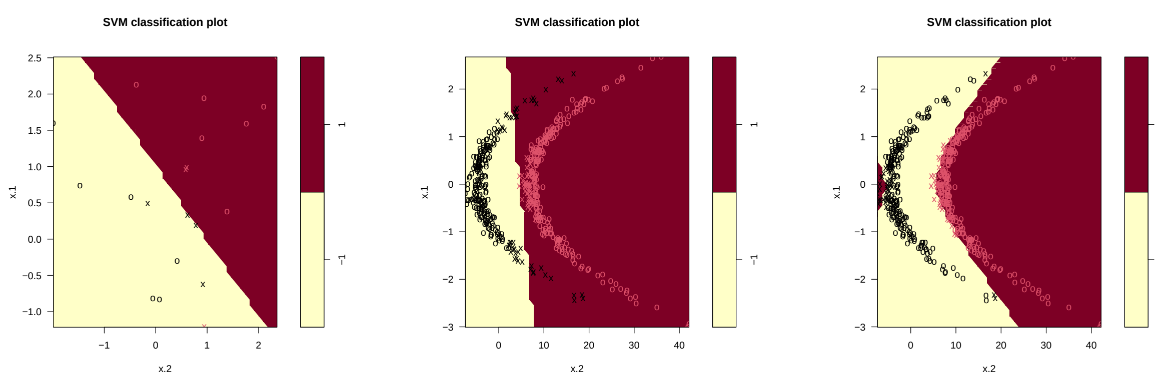
(1) Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.
   - Ch. 16.2 Support Vector Machine

(2) Bishop, C. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: Springer.
   - Ch. 6.1, 6.2 Kernel methods

(3) Vapnik, V. (2013). The nature of statistical learning theory. Springer science & business media.

1. Intro and motivation

Note 1. Consider the Soft SVM with predictive rule $h(x) = \text{sign}(\eta(x))$ with separator $\eta(x) = \langle w, x\rangle + b$ where $w = (w_1, w_2)^\top \in \mathbb{R}^2$ and $x = (x_1, x_2)^\top \in \mathbb{R}^2$. It can address learning problems where the data can (up to some degree of violation) be separated by a line (Figure 1.1a). In more challenging cases where the geometry is strongly non-linear this can totally fail (Figure 1.1b).



(a) Linearly separable data set, with linear separator

(b) Non-linearly separable data set, with linear separator

(c) Non-linearly separable data set, with quadratic separator

Figure 1.1. Soft SVM from Computer practical 4

Note 2. The accuracy of predictive rule could be improved if I could take into account the curvature by adding a quadratic term in the 2nd dimension as $h^{\psi}(x) = \text{sign}\left(\eta^{\psi}(x)\right)$ with separator $\eta^{\psi}(x) = \langle w', \psi(x)\rangle + b'$ for some $\psi : \mathbb{R}^2 \to \mathbb{R}^3$ with $\psi(x) = \left(x_1, x_2, x_2^2\right)$ and learning w' and b' . It works, (Figure 1.1c).

Note 3. In other words, in order to improve the expressiveness of a given hypothesis class $\mathcal{H} = \{x \mapsto f(\langle w, x\rangle)\}$ (for some function f) with purpose to learn a more accurate predictive rule, it is reasonable to consider an embedding $\psi(x)$ and work on the learning problem with $\mathcal{H} = \{x \mapsto f(\langle w, \psi(x)\rangle)\}$. Such an embedding $\psi(x)$ can possibly be a vector of basis functions such as polynomials, splines, etc...

Note 4. The above may drastically increase the dimensionality of the problem hence the computational cost and required training dataset size. This challenge is addressed by the Kernel trick which allows the design of cheap more expressive extensions of many well known algorithms.

2. Improving expressive power via embeddings in feature spaces

Note 5. To make the class of hypotheses more expressive with purpose to improve accuracy, we can first map the original instance space $x \in \mathcal{X}$ into another feature space $\mathcal{F}$ (possibly of a higher dimension) via an embedding $\psi : \mathcal{X} \to \mathcal{F}$ and then learn a hypothesis in that space.

Note 6. Consider a given learning task that involves a hypothesis class $\mathcal{H} = \{x \mapsto \langle w, x\rangle : w \in \mathbb{R}^n\}$ where the predictive rule $h \in \mathcal{H}$ is defined over domain set $\mathcal{X}$ and is to be trained against data set $\mathcal{S} = \{z_i = (x_i, y_i)\}_{i=1}^m$ drawn from data generating process $G(\cdot)$. The basic paradigm for improving expressive power of $\mathcal{H}$ via embedding involves:

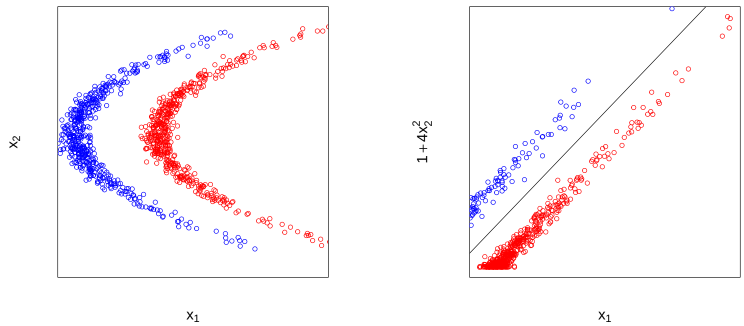
(1) Choose an embedding $\psi : \mathcal{X} \to \mathcal{F}$ with $\psi(x) := (\psi_1(x), ..., \psi_d(x))^\top$ for some feature space $\mathcal{F}$.
(2) Create the image sequence $\mathcal{S}^{\psi} = \left\{z_i^{\psi} = (\psi(x_i), y_i)\right\}_{i=1}^m$ from the original training set $\mathcal{S}$.
(3) Train a linear predictor h against $\mathcal{S}^{\psi}$.
(4) Predict the label or the output of a new point x^{new} by $h^{\psi}(x^{\text{new}}) := h \circ \psi(x^{\text{new}}) = h(\psi(x^{\text{new}}))$

Note 7. The introduction of embedding $\psi : \mathcal{X} \to \mathcal{F}$ induces

(1) a probability distribution G^{ψ} over domain $\mathcal{F} \times \mathcal{Y}$ with $G^{\psi}(A) = G\left(\psi^{-1}(A)\right)$ for every set $A \subseteq \mathcal{F} \times \mathcal{Y}$ given a data generating process $G(\cdot)$.
(2) predictive rule $h^{\psi}(\cdot) := h \circ \psi(\cdot) = h(\psi(\cdot))$
(3) risk function $R_{G^{\psi}}(h) := R_G(h \circ \psi)$, as

$$R_G(h \circ \psi) = \int \ell(h \circ \psi, z = (x, y))\,\mathrm{d}G(z) = \int \ell\left(h, z^{\psi}\right)\mathrm{d}G^{\psi}(x, y) = R_{G^{\psi}}(h)$$

Note 8. For a specific learning task, the success of the learning paradigm in Note 6 depends on choosing an embedding ψ that provides a suitable deformation for the image of the data generating process (or the training dataset) to be as close as possible to what could be accurately addressed by the specific learning task. Eg, in SVM, ψ will make the image of the data distribution (close to being) linearly separable in the feature space $\mathcal{F}$, thus making the resulting learning algorithm a



(A) Inputs in the original space x_1 vs x_2

(B) Projection of inputs tot he feature space $\psi_1(x) = x_1$ vs $\psi_2(x) = 4x_2^2+1$

FIGURE 2.1. Projection of the inputs living in the original space to the feature space

good learner for a given task (Figure 2.1). This requires prior knowledge of the problem (In Section 4, we see popular recipes for that).

Note 9. As feature mapping ψ any function that maps the original instances $\mathcal{X}$ into some Hilbert space $\mathcal{F}$ can be used.

Definition 10. A Hilbert space is a vector space, with an inner product, which is also complete.

Lemma 11. *If $\mathcal{X}$ is a linear subspace of a Hilbert space, then every $x \in \mathcal{X}$ can be written as $x = u + v$ where $u \in \mathcal{X}$ and $\langle u, v\rangle = 0$ for all $v \in \mathcal{X}$.*

Note 12. Feature space $\mathcal{F}$ is a Hilbert space preferably due to Lemma 11 that enables the Kernel trick via the representation Theorem 19. Eg, a Euclidean space such as $\mathbb{R}^d$ for some d. That includes infinite dimensional spaces.

Note 13. Using a $\psi(x) = (\psi_1(x), \psi_2(x) ...)^\top$ that is high dimensional (d is too large) may improve accuracy (expressiveness) of the learner (e.g. recall in polynomial regression increasing the polynomial degree). However this increases the computational effort/cost required to perform calculations to minimize the associated risk function in the high dimensional space, as well as we need more data. This is addressed via the Kernel trick.

Example 14. ; Example Consider the Soft-SVM in "Handout 7: Support Vector Machines". Consider a given embedding $\psi : \mathcal{X} \to \mathcal{F}$ with $\psi(x) = (\psi_1(x), \psi_2(x) ...)^\top$. The learning rule becomes

$$h^{\psi}(x) = h(\psi(x)) = \text{sign}(\langle w, \psi(x)\rangle + b) \tag{2.1}$$

. In "Handout 7, Problem 26 becomes

Solve

$$(w^*, b^*, \xi^*) = \underset{(w,b,\xi)}{\arg\min} \left(\lambda \|w\|_2^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right)$$

$$\text{subject to: } y_i \left(\langle w, \psi(x_i) \rangle + b \right) \geq 1 - \xi_i, \ \forall i = 1, ..., m$$

$$\xi_i \geq 0, \ \forall i = 1, ..., m$$

3. The Kernel trick

Note 15. We discuss a duality in the learning problem 16 that facilitates the implementation of the extension to a possibly high dimensional feature space (hence improving the expressiveness/accuracy) by using kernel functions (hence reducing dimensionality, computational cost, and required data size).

Problem 16. (Learning problem) Consider a prediction rule $h : \mathcal{X} \to \mathcal{Y}$ with $h(x) = \langle w, \psi(x) \rangle$ which is trained against a training sample $\{z_i = (x_i, y_i)\}_{i=1}^m$ with the following general optimization problem

$$\underset{w}{\text{minimize}} \left(\ f \left(\langle w, \psi(x_1) \rangle, ..., \langle w, \psi(x_m) \rangle \right) \ + \ R(\|w\|) \ \right), \tag{3.1}$$

where $f : \mathbb{R}^m \to \mathbb{R}$ is an arbitrary function and $R : \mathbb{R}_+ \to \mathbb{R}$ is a monotonically non-decreasing function.

Example 17. (Cont. Example 14) Soft SVM problem has a solution equivalent to (see Proposition 31, Handout 7)

$$(w^*, b^*) = \underset{(w,b)}{\arg\min} \left(\frac{1}{m} \sum_{i=1}^{m} \max\left(0, 1 - y_i \left(\langle w, \psi(x_i) \rangle + b \right)\right) + \lambda \|w\|_2^2 \right)$$

then in terms of (3.1) we get

$$f(\alpha_1, ..., \alpha_m) = \frac{1}{m} \sum_{i=1}^{m} \max\{0, 1 - y_i \alpha_i\}, \text{ and } R(\beta) = \lambda \beta^2$$

Definition 18. Kernel function K is defined as $K : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ with $K(x, x') = \langle \psi(x), \psi(x') \rangle$ given an embedding $\psi(x)$ of some domain space $\mathcal{X}$ into some Hilbert space $\mathcal{F}$. Kernel functions describe inner products in the feature space $\mathcal{F}$.

Theorem 19. *(Representation theorem) Assume mapping $\psi : \mathcal{X} \to \mathcal{F}$ where $\mathcal{F}$ is a Hilbert space. There exists a vector $\alpha \in \mathbb{R}^m$ such that $w = \sum_{i=1}^m \alpha_i \psi(x_i)$ is the optimal solution of (3.1) in Problem 16.*

Proof. Let w^* be the optimal solution of (3.1). Because w^* is element of Hilbert space, it can be written as $w^* = \sum_{i=1}^m \alpha_i \psi(x_i) + u$ where $\langle u, \psi(x_i) \rangle = 0$ for all $i = 1, ..., m$. Set $w := w^* - u$.

Because $\|w^*\|^2 = \|w\|^2 + \|u\|^2$ it is $\|w\| \leq \|w^*\|$ implying that

$$R(\|w\|) \leq R(\|w^*\|).$$

Because $\langle w, \psi(x_i)\rangle = \langle w^* - u, \psi(x_i)\rangle = \langle w^*, \psi(x_i)\rangle$ for all $i = 1, ..., m$, it is

$$f(\langle w, \psi(x_1)\rangle, ..., \langle w, \psi(x_m)\rangle) = f(\langle w^*, \psi(x_1)\rangle, ..., \langle w^*, \psi(x_m)\rangle)$$

Then the objective function of (3.1) at w is less than or equal to that of the minimizer w^* which implies that $w = \sum_{i=1}^{m} \alpha_i \psi(x_i)$ is an optimal solution. □

Note 20. Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function with $K(x, x') = \langle \psi(x), \psi(x')\rangle$. According to the representation Theorem 19, the Learning problem 16, can be equivalently addressed by re-writing the learning predictive rule as

$$h_\alpha(x) = \sum_{i=1}^{m} \alpha_i K(x_i, x)$$

and learning $\{\alpha_i\}$ as the solutions of

$$\text{(3.2)} \quad \underset{\alpha}{\text{minimize}} \left(f\left(\sum_{i=1}^{m} \alpha_i K(x_i, x_1), ..., \sum_{i=1}^{m} \alpha_i K(x_i, x_m)\right) + R\left(\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j K(x_i, x_j)}\right)\right),$$

This is because

$$\langle w, \psi(x_j)\rangle = \langle \sum_{i=1}^{m} \alpha_i \psi(x_i), \psi(x_j)\rangle = \sum_{i=1}^{m} \alpha_i \langle \psi(x_i), \psi(x_j)\rangle = \sum_{i=1}^{m} \alpha_i K(x_i, x_j)$$

and

$$\|w\|^2 = \langle \sum_{i=1}^{m} \alpha_i \psi(x_i), \sum_{j=1}^{m} \alpha_j \psi(x_j)\rangle = \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j \langle \psi(x_i), \psi(x_j)\rangle = \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j K(x_i, x_j)$$

Note 21. In Learning Problem 16, direct access to elements $\psi(\cdot)$ in the feature space is not necessary because equivalently one can calculate or just specify the associated kernel function (that is inner products in the feature space).

Example 22. (Cont. Example 17) In Soft SVM the predictive rule becomes $h(x) = \text{sign}\left(\sum_j \alpha_j K(x_i, x_j)\right)$ and the learning task becomes

$$\underset{\alpha}{\text{minimize}} \left(\lambda \sum_i \sum_j \alpha_i \alpha_j K(x_i, x_j) + \frac{1}{m} \sum_i \max\left(0, 1 - y_i \sum_j \alpha_j K(x_i, x_j)\right)\right)$$

which can be addressed via SGD. This form of SVM is called Kernel SVM because we can just directly specify the kernel function $K(\cdot, \cdot)$ without the need to even think about feature mapping $\psi(\cdot)$ (which is eliminated and replaced by the kernel).

Example 23. (Polynomial Kernels) Let $x \in \mathcal{X} \subseteq \mathbb{R}^n$. Assume we want to extend the linear mapping $x \mapsto \langle w, x\rangle$ to the k degree polynomial mapping $x \mapsto h(x)$. The multivariate polynomial can be written as $h(x) = \langle w, \psi(x)\rangle$, where $\psi : \mathbb{R}^n \to \mathbb{R}^d$ with $\psi(x)$ is a vector of elements $\psi_J(x) = \prod_{i=1}^{r} x_{J_i}$ for $J \in \{1, ..., n\}^r$ and $r \leq k$. This learning problem can be equivalently be addressed with the k degree polynomial kernel

$$K(x, x') = (1 + \langle x, x'\rangle)^k$$

Solution. It is

$$\begin{aligned}
K\left(x,x'\right) &= \left(1+\langle x,x'\rangle\right)^k = \left(\sum_{j=0}^{n} x_j x'_j\right)^k \quad \text{(by setting } x_0 = x'_0 = 1\text{)}\\
&= \sum_{J\in\{0,1,\ldots,n\}^k} \prod_{i=1}^{k} x_{J_i} x'_{J_i} = \sum_{J\in\{0,1,\ldots,n\}^k} \left(\prod_{i=1}^{k} x_{J_i}\right)\left(\prod_{i=1}^{k} x'_{J_i}\right)\\
&= \langle \psi\left(x\right), \psi\left(x'\right)\rangle
\end{aligned}$$

where $\psi\left(x\right)$ is as defined.

Example 24. (Radial basis kernel) Let the original input space be $x \in \mathcal{X} \subseteq \mathbb{R}$. Consider the Radial Basis Functions Kernel (or Gaussian kernel)

$$K\left(x,x'\right) = \exp\left(-\frac{1}{2\sigma^2}\left\|x-x'\right\|_2^2\right).$$

Show that it is a kernel indeed, by presenting it as an inner product in a feature space of infinite dimension, and state the bases of the mapping $\psi\left(\cdot\right)$.

Solution. It is

$$\begin{aligned}
K\left(x,x'\right) =& \exp\left(-\frac{1}{2\sigma^2}\left\|x-x'\right\|_2^2\right) = \exp\left(\frac{1}{\sigma^2}xx' - \frac{1}{2}x^2 - \frac{1}{2}\left(x'\right)^2\right)\\
=& \exp\left(\frac{1}{\sigma^2}xx'\right)\exp\left(-\frac{1}{2\sigma^2}x^2\right)\exp\left(-\frac{1}{2\sigma^2}\left(x'\right)^2\right)\\
=& \sum_{k=0}^{\infty}\frac{\left(xx'/\sigma^2\right)^k}{k!}\exp\left(-\frac{1}{2\sigma^2}x^2\right)\exp\left(-\frac{1}{2\sigma^2}\left(x'\right)^2\right) \quad \text{(...by Taylor Expansion)}\\
=& \sum_{k=0}^{\infty}\left[\frac{x^k}{\sqrt{k!}\sigma^k}\exp\left(-\frac{1}{2\sigma^2}x^2\right)\right]\left[\frac{\left(x'\right)^k}{\sqrt{k!}\sigma^k}\exp\left(-\frac{1}{2\sigma^2}\left(x'\right)^2\right)\right]
\end{aligned}$$

hence it is $K\left(x,x'\right) = \langle\psi\left(x\right),\psi\left(x'\right)\rangle$ with $\psi_k\left(x\right) = \frac{x^k}{\sqrt{k!}\sigma^k}\exp\left(-\frac{1}{2\sigma^2}x^2\right)$.

4. Construction of kernels

Note 25. The kernel formulated as an inner product in a feature space allows us to build interesting extensions of many well-known algorithms by making use of the kernel trick and without the need to have direct access to the feature space (E.g. Example 22).

Note 26. Specifying a kernel function K is a way to express prior knowledge without the need to have direct access to the feature space. This is consequence of the Representation theorem 19 that kernel is the inner product of feature mappings ψ which sufficiently replaces them in the learning problem, and the fact that ψ is a way to express and utilize prior knowledge about the problem at hand.

Note 27. Theorem 30 provides sufficient and necessary conditions to check whether the specified function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is indeed a kernel function; i.e. if K can be written as inner product $K(x, x') = \langle \psi(x), \psi(x') \rangle$ of feature functions $\psi(x)$.

Definition 28. Gram matrix is called the $m \times m$ matrix G s.t. $[G]_{i,j} = K(x_i, x_j)$.

Definition 29. A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive semi-definite if its Gram matrix G, $[G]_{i,j} = K(x_i, x_j)$, is a positive semi-definite matrix.

Theorem 30. *A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ implements an inner product in some Hilbert space is a valid kernel function if and only if it is positive semi-definite i.e. its Gram matrix G , $[G]_{i,j} = K(x_i, x_j)$, is a positive semi-definite matrix.*

Proof. Assume K is a valid kernel function (i.e. it implements an inner product in some Hilbert space) $K(x, x') = \langle \psi(x), \psi(x') \rangle$; let's consider $\psi : \mathcal{X} \to \mathbb{R}^d$ for simplicity. Let G be its Gram matrix with $G = \Psi^\top \Psi$ and $\psi(x_i)$ is the i-th column of Ψ. For any $\xi \in \mathbb{R}^d - \{0\}$

$$\xi^\top G \xi = \sum_i \sum_j \xi_i K(x_i, x_j) \xi_j = \sum_i \sum_j \xi_i \langle \psi(x_i), \psi(x_j) \rangle \xi_j = \sum_i \sum_j \langle \xi_i \psi(x_i), \psi(x_j) \xi_j \rangle$$

$$= \langle \sum_i \xi_i \psi(x_i), \sum_j \psi(x_j) \xi_j \rangle = \left\| \sum_i \xi_i \psi(x_i) \right\|_2^2 \geq 0$$

Assume the symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive semi-definite. Let $\mathbb{R}^f = \{f : \mathcal{X} \to \mathbb{R}\}$. For $x \in \mathcal{X}$ let function ψ over $\mathbb{R}^f$ with $\psi(x) = K(\cdot, x)$. This allows to define a vector space consisting of all the linear combinations of elements of the form $K(\cdot, x)$, having an inner product

$$\langle \sum_i \alpha_i K(\cdot, x_i), \sum_j \beta_j K(\cdot, x_j) \rangle = \sum_i \sum_i \alpha_i \beta_j \underbrace{\langle K(\cdot, x_i), K(\cdot, x_j) \rangle}_{=K(x_i, x_j)}.$$

This is satisfies all the properties of inner product, s.t. it is symmetric, linearity, positive definite as $K(x, x') \geq 0$. Then there is some feature vector ψ such that $K(x, x') = \langle \psi(x), \psi(x') \rangle$. □

Claim 31. In Figure 1.1c, we could see that examples/points can be distinguished by some ellipse, so it was reasonable for the scientist to define ψ as a vector with elements all the monomials up to order ; alternatively we could use a degree 2 polynomial kernel.

Proposition 32. *A powerful technique for constructing new kernels is to build them out of simpler kernels as building blocks. Below are some properties. Assume $K_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $K_2 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ are valid kernels, then the following are kernels too*

(1) $K(x, x') = K_1(x, x') + K_2(x, x')$
(2) $K(x, x') = K_1(x, x') K_2(x, x')$
(3) $K(x, x') = K_1(x_1, x_1') + K_2(x_2, x_2')$, *where* $x = (x_1, x_2)^\top$, $x' = (x_1', x_2')^\top$
(4) $K(x, x') = K_1(x_1, x_1') K_2(x_2, x_2')$, *where* $x = (x_1, x_2)^\top$, $x' = (x_1', x_2')^\top$
(5) $K(x, x') = f(x) K_1(x, x') f(x')$ *for any function* f
(6) $K(x, x') = K_1(f(x), f(x'))$ *for any function* f

Solution. We present the first two and the rest are proved similarly.

See the proof as a solution to an Example

For (1). Let Gram matrix, G_j induced by kernel function K_j. For any $\xi \in \mathbb{R}^d - \{0\}$

$$\xi^\top G_3 \xi = \xi^\top (G_1 + G_2) \xi = \xi^\top G_1 \xi + \xi^\top G_2 \xi \geq 0$$

For (2). Assume that $K_j(x, x') = (\psi_j(x))^\top \psi_j(x)$. Then

$$\begin{aligned} K\left(x, x'\right) &= K_1\left(x, x'\right) K_2\left(x, x'\right) = (\psi_1(x))^\top \psi_1\left(x'\right) (\psi_2(x))^\top \psi_2\left(x'\right) \\ &= (\psi_1(x))^\top \psi_2(x) \left(\psi_1\left(x'\right)\right)^\top \psi_2\left(x'\right) = \left((\psi_1(x))^\top \psi_2(x)\right)^\top \left(\psi_1\left(x'\right)\right)^\top \psi_2\left(x'\right) \end{aligned}$$

which can be represented as an inner product of feature vectors.

Note 33. The concept of a kernel formulated as an inner product in a feature space allows us to build interesting extensions of many well-known algorithms by making use of the kernel trick. One example was the Kernel SVM. Some other popular cases are Gaussian process regression, and Kernel PCA.