# STAT37710/CMSC35400 — Homework 4

University of Chicago, Spring 2018 — due Tuesday, May 15 at 11:59pm

1. Prove that a symmetric matrix $K \in \mathbb{R}^{d\times d}$ is positive semi-definite if and only if its eigenvalues are nonnegative.

2. Give an example of function $k\colon \mathcal{X}\times\mathcal{X}\to\mathbb{R}$ that is symmetric ($k(x,x') = k(x',x)$) and positive in the sense that $k(x,x')\geq 0$ for all $x,x'\in\mathcal{X}$, but is not positive semidefinite. Conversely, give an example of a kernel that is positive semidefinite, but does not satisfy $k(x,x')\geq 0$ for all $x,x'\in\mathcal{X}$.

3. Given any function $\psi\colon \mathcal{X}\to\mathcal{X}'$, prove that if $k'$ is a psd kernel on $\mathcal{X}'$, then $k(x,x') = k'(\psi(x),\psi(x'))$ is a psd kernel on $\mathcal{X}$.

4. Prove that if $k_1$ and $k_2$ are two positive semi-definite (psd) kernels on a space $\mathcal{X}$, then

   (a) the function $k(x,x') := k_1(x,x') + k_2(x,x')$ is a psd kernel on $\mathcal{X}$;

   (b) The function $k_{\oplus}((x_1,x_2),(x_1',x_2')) = k_1(x_1,x_1') + k_2(x_2,x_2')$ is a psd kernel on $\mathcal{X}\times\mathcal{X}$.

   (c) Let $\alpha(\mathbf{x},\mathbf{x}')$ be the angle between $\mathbf{x},\mathbf{x}'\in\mathbb{R}^n$. Prove that the cosine kernel $k_{\angle}(\mathbf{x},\mathbf{x}') = \cos\alpha(\mathbf{x},\mathbf{x}')$ is a psd kernel on $\mathcal{X} = \mathbb{R}^n$.

5. In machine learning one often encounters problems that are invariant to some symmetry group. For example, in learning to recognize objects, if $h(x)$ is the predicted label of object $x$ and $T_\alpha(x)$ is the same object rotated by $\alpha$ degrees, one would like $h$ to satisfy $h(T_\alpha(x)) = h(x)$ for any $\alpha$. Formally, this situation corresponds to considering a collection $\mathcal{T}_G = \{\, T_g \mid g\in G\,\}$ of transformations $T_g\colon \mathcal{X}\to\mathcal{X}$ parametrized by the symmetry group $G$. For simplicity, let us assume that $G$ is finite.

   Given a kernel $k\colon \mathcal{X}\times\mathcal{X}\to\mathbb{R}$, the "best alignment" similarity measure

   $$k_{max}(x,x') = \max_{g,g'\in G} k(T_g(x),T_{g'}(x'))$$

   is certainly invariant to $\mathcal{T}_G$. Show by providing a counterexample that $k_{max}$ is not necessarily positive semi-definite, though. In contrast, show that

   $$k_G(x,x') = \sum_{g,g'\in G} k(T_g(x),T_{g'}(x'))$$

   is both invariant and positive semi-definite. Show that the output of any kernel based learning algorithm defined in terms of this kernel will satisfy $h(T_g(x)) = h(x)$ for any $g\in G$. Describe the RKHS and feature map corresponding to $k_G$. If you are into abstract algebra, further show that provided that $\mathcal{T}_G$ is itself a group (as it usually is), then

   $$k_G'(x,x') = \sum_{g,g'\in G} k(T_g(x),x')$$

   is also an invariant semi-definite kernel and has essentially the same RKHS as $k_G$.

6. Given a finite alphabet $\Sigma$ (e.g., $\Sigma = \{a,b,\ldots,z\}$), $\overline{x}\in\Sigma^\ell$ denotes that $\overline{x} = x_0x_2\ldots x_{\ell_{\overline{x}}-1}$ is a string of length $\ell_{\overline{x}}$, in which each character $x_i$ comes from $\Sigma$. For any increasing sequence $0\leq i_1\leq\ldots\leq i_k\leq \ell_{\overline{x}-1}$, we will use $\overline{x}_{(i_1,\ldots,i_k)}$ to denote the string $x_{i_1}x_{i_2}\ldots x_{i_k}$, and we will use $\overline{x}_{i:j}$ as a shorthand for $x_ix_{i+1}\ldots x_j$. Given two characters $u$ and $v$, we will also use the notation

   $$[\![u=v]\!] = \begin{cases} 1 & \text{if } u=v \\ 0 & \text{if } u\neq v. \end{cases}$$

(a) Let $n_{\overline{s}}(\overline{x})$ be the number of places that $\overline{s}$ appears in $\overline{x}$ as a contiguous substring, i.e., $n_{\overline{s}}(\overline{x}) = |\{\, i \mid 0 \le i \le \ell_{\overline{x}} - \ell_{\overline{s}},\ \overline{x}_{i:i+\ell_{\overline{s}}-1} \,\}|$. Define the $\kappa$'th order contiguous substring kernel between two strings $\overline{a}$ and $\overline{b}$ as

$$k_\kappa(\overline{a}, \overline{b}) = \sum_{\overline{s} \in \Sigma^\kappa} n_{\overline{s}}(\overline{a})\, n_{\overline{s}}(\overline{b}).$$

Explain the intuitive meaning of this kernel and why it is positive semi-definite. Write down the pseudocode of an $O(\kappa \ell_{\overline{a}} \ell_{\overline{b}})$ running time algorithm to compute $k_\kappa(\overline{a}, \overline{b})$ (here, one unit of computation is testing a single character $z$ against a specific character $a_i$ of $\overline{a}$ or $b_j$ of $\overline{b}$).

(b) Let $n^*_{\overline{s}}(\overline{x})$ be the number of places that $\overline{s}$ appears in $\overline{x}$ as a gappy substring, i.e.,

$$n^*_{\overline{s}}(\overline{x}) = \Big|\, \{\, (i_1, \ldots, i_{\ell_{\overline{s}}}) \mid 0 \le i_1 < i_2 < \ldots < i_{\ell_{\overline{s}}} \le \ell_{\overline{x}} - 1,\ \overline{x}_{(i_1, i_2, \ldots, i_{\ell_{\overline{s}}})} = \overline{s} \,\} \,\Big|.$$

Define the $\kappa$'th order gappy substring kernel between two strings $\overline{a}$ and $\overline{b}$ as

$$k^*_\kappa(\overline{a}, \overline{b}) = \sum_{\overline{s} \in \Sigma^\kappa} n^*_{\overline{s}}(\overline{a})\, n^*_{\overline{s}}(\overline{b}). \tag{1}$$

Explain the intuitive meaning of this kernel and why it is positive semi-definite.

i. Devise an $O(|\Sigma| + \ell_{\overline{a}} + \ell_{\overline{b}})$ algorithm for computing $k^*_1(\overline{a}, \overline{b})$ and write down its pseudocode.

ii. Let $1 \le p \le \kappa$. Devise an $O(p)$ algorithm for computing $k^*_p(\overline{a}_{0:p-1}, \overline{b}_{0:p-1})$.

iii. Assume that $k^*_{p-1}(\overline{a}_{0:i}, \overline{b}_{0:j})$ and $k^*_p(\overline{a}_{0:i}, \overline{b}_{0:j})$ have been computed for all $0 \le i \le N$ and $0 \le j \le M$ for some $(N, M)$.

A. For any fixed $j \in \{0, \ldots, M\}$, derive a formula for $k^*_p(\overline{a}_{0:N+1}, \overline{b}_{0:j})$ involving only already computed quantities plus the indicators $[\![a_{N+1} = b_{p-1}]\!], \ldots, [\![a_{N+1} = b_j]\!]$. Please don't just state the final formula, but also justify it.

B. For any fixed $i \in \{0, \ldots, N\}$, derive a formula for $k^*_p(\overline{a}_{0:i}, \overline{b}_{0:M+1})$ involving only already computed quantities plus the indicators $[\![a_{p-1} = b_{M+1}]\!], \ldots [\![a_i = b_{M+1}]\!]$. Please don't just state the final formula, but also justify it.

C. Combining parts (i)–(iii), derive an $O(\kappa \ell_{\overline{a}} \ell_{\overline{b}})$ algorithm to compute $k^*_\kappa(\overline{a}, \overline{b})$ and write down its pseudocode.

(c) A potential drawback of the gappy substring kernel is that the contribution of any gappy match is counted the same, regardless of whether $\overline{s}$ occurs as a contiguous substring of $\overline{a}$ and $\overline{b}$ or whether it has huge gaps. One solution that has been proposed to this problem is to penalize the match by the number of skipped characters by defining

$$w_{\overline{s}}(\overline{x}) = \sum_{0 \le i_1 < i_2 < \ldots < i_{\ell_{\overline{s}}} \le \ell_{\overline{x}} - 1} \lambda^{i_{\ell_{\overline{s}}} - i_1 - (\ell_{\overline{s}} - 1)}\, [\![\, \overline{x}_{(i_1, i_2, \ldots, i_{\ell_{\overline{s}}})} = \overline{s} \,]\!]$$

for some penalty factor $\lambda < 1$, and defining the kernel as

$$k^\square_\kappa(\overline{a}, \overline{b}) = \sum_{\overline{s} \in \Sigma^\kappa} w_{\overline{s}}(\overline{a})\, w_{\overline{s}}(\overline{b}).$$

Repeat part (b) of this question for $k^\square_\kappa$ (part (iii) will have to be somewhat modified) and show that $k^\square_\kappa$ can also be computed in time $O(\kappa \ell_{\overline{a}} \ell_{\overline{b}})$.

(d) **(extra credit)** Another useful variation on the gappy substring kernel is to define a kernel $k_\Sigma$ on the alphabet itself, and, instead of counting exact (gappy) matches between $\overline{a}$ and $\overline{b}$, define the kernel as

$$k^\triangle_\kappa(\overline{a}, \overline{b}) = \sum_{0 \le i_1 < i_2 < \ldots < i_\kappa \le \ell_{\overline{a}} - 1} \sum_{0 \le j_1 < j_2 < \ldots < j_\kappa \le \ell_{\overline{b}} - 1} \prod_{t=1}^{\kappa} k_\Sigma(a_{i_t}, b_{j_t}).$$

Describe the intuitive meaning of this kernel, why it is positive semi-definite, and show that this kernel can also be computed in time $O(\kappa \ell_{\overline{a}} \ell_{\overline{b}})$.