

Homework 3

CS229T/STATS231 (Winter 2014–2015)

Please structure your writeups hierarchically: convey the overall plan before diving into details. You should justify with words why something's true (by algebra, convexity, etc.). There's no need to step through a long sequence of trivial algebraic operations. Be careful not to mix assumptions with things which are derived. Up to two additional points will be awarded for especially well-organized and elegant solutions.

Due date: Wednesday, March 4 (at the beginning of class)

1. Kernels (15 points) This problem will explore a number of kernels and non-kernels to get some more intuition for (i) what constitutes a valid kernel and (ii) what kind of functions we can implicitly define with kernels.

(a) Let the input space be $\mathcal{X} = \mathbb{R}^d$. Define the kernel to be 1 for closeby points and 0 otherwise:

$$k(x, x') = \begin{cases} 1 & \text{if } \|x - x'\|_2 \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

Is k a valid kernel? Justify your result either way.

(b) Suppose k' is a valid kernel. Define the normalized kernel k based on k' as follows:

$$k(x, x') = \frac{k'(x, x')}{\sqrt{k'(x, x)k'(x', x')}}.$$

Is k a valid kernel? Justify your result either way.

(c) Suppose $k' : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ is a valid kernel, and let $\mathcal{F}$ be a finite collection of functions from $\mathcal{X}$ to $\mathcal{Y}$. Define k as follows:

$$k(x, x') = \sum_{f \in \mathcal{F}} \sum_{f' \in \mathcal{F}} k'(f(x), f'(x')).$$

Is k a valid kernel? Justify your result either way.

(d) Let $\mathcal{X} = [0, 1]$. Define:

$$k(x, x') = \min(x, x').$$

Show that k is a valid kernel. Hint: given n points $x_1, \dots, x_n$, try to construct a n -dimensional multivariate Gaussian distribution with covariance matrix equal to the kernel matrix.

It turns out that k is the reproducing kernel for an RKHS over the space of functions differentiable almost everywhere:

$$\mathcal{H} = \{f : f(0) = 0 \text{ and } f' \in L^2([0, 1])\},$$

with $\langle f, g \rangle = \int_0^1 f'(x)g'(x)dx$. Show the reproducing property for k holds (that $\langle k(\cdot, x), f \rangle = f(x)$).

2. A factorization trick for kernelized classification (5 points)

Recall the online learning multiclass classification loss function from homework 1:

$$\ell_t(w) = \max_{y \in \mathcal{Y}} \left\{ \ell(y^{(t)}, y) + \left\langle w_y, \phi(x^{(t)}) \right\rangle - \left\langle w_{y^{(t)}}, \phi(x^{(t)}) \right\rangle \right\}. \quad (1)$$

Before, the feature map ϕ was the identity function. In this assignment, we will explore how to predict new examples when ϕ is a more complicated function.

Recall that for label $y \in \mathcal{Y}$, we can always express the weights after seeing n examples as $w_y = \sum_{i=1}^n \alpha_{iy} \phi(x_i)$, yielding the prediction function:

$$f_y(x_j) = \langle w_y, \phi(x_j) \rangle = \sum_{i=1}^n \alpha_{iy} k(x_i, x_j).$$

Define the kernel matrix to be K , where $K_{ij} = k(x_i, x_j)$.

Suppose you've already written code that works with finite-dimensional feature vectors and weight vectors, but now you want to use kernels without changing your code. Given a kernel, we will construct a feature space so that doing ordinary learning in the new feature space corresponds to using that kernel.

The key will be to obtain a factorization of the kernel matrix $K = LL^T$ (e.g., Cholesky). We can then interpret L as the feature matrix (where the i -th row is a new feature vector of x_i).

- (a) Show that for any α and L , we can define some $v_y \in \mathbb{R}^n$ in terms of α and L such that

$$f_y(x_j) = \langle v_y, L_j \rangle,$$

where $L_j \in \mathbb{R}^n$ is the j -th row of L . Note that we can interpret v_y as just a weight vector and L_j is the feature vector, both n -dimensional. The implication is that given our data and kernel, we can factorize K to get L , and then run any learning algorithm that takes feature vectors to tune v_y .

- (b) After fitting $v_y \in \mathbb{R}^n$, we would like to use it to predict on new examples. Show that you can do some offline preprocessing so that you can compute $f_y(x)$ in $O(n)$ time (you can assume that a kernel evaluation is $O(1)$ time).

3. Feedback (0 points)

- (a) On a scale of 1 to 10, how difficult was this assignment?
- (b) On a scale of 1 to 10, how useful was this assignment?
- (c) Any other comments?