

# Assignment 3: Kernel Methods

CS489/698 – Winter 2018

Out: February 12, 2018

Due: March 2 (11:59pm)

**Submit an electronic copy of your assignment via LEARN. Late submissions incur a 2% penalty for every rounded up hour past the deadline. For example, an assignment submitted 5 hours and 15 min late will receive a penalty of  $\text{ceiling}(5.25) * 2\% = 12\%$ .**

**Be sure to include your name and student number with your assignment.**

1. **[20 pts]** Show that the Gaussian kernel  $k(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2)$  can be expressed as the inner product of an infinite-dimensional feature space. Hint: use the following expansion and show that the middle factor further expands as a power series:

$$k(\mathbf{x}, \mathbf{x}') = e^{-\mathbf{x}^T \mathbf{x} / 2\sigma^2} e^{\mathbf{x}^T \mathbf{x}' / \sigma^2} e^{-(\mathbf{x}')^T \mathbf{x}' / 2\sigma^2}$$

2. **[30 pts]** For this question, you will develop a dual formulation of the perceptron learning algorithm. Using the perceptron learning rule

$$\mathbf{w}^{t+1} = \begin{cases} \mathbf{w}^t + y_n \phi(\mathbf{x}_n) & \text{if } y_n \mathbf{w}^T \phi(\mathbf{x}_n) \leq 0 \\ \mathbf{w}^t & \text{otherwise} \end{cases}$$

show that the learned weight vector  $\mathbf{w}$  can be written as a linear combination of the vectors  $y_n \phi(\mathbf{x}_n)$  where  $y_n \in \{-1, +1\}$ . Denote the coefficients of this linear combination by  $a_n$ .

- (a) **[15 pts]** Derive a formulation of the perceptron learning rule in terms of  $a_n$ . Show that the feature vector  $\phi(\mathbf{x})$  appears in the perceptron learning rule only as part of the kernel function  $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$ .
- (b) **[15 pts]** Derive a formulation of the predictive learning rule

$$y = \begin{cases} 1 & \text{if } \mathbf{w}^T \phi(\mathbf{x}) > 0 \\ -1 & \text{otherwise} \end{cases}$$

in terms of  $a_n$ .

3. **[50 pts]** Non-linear regression techniques.

Implement the following regression algorithms. A dataset will be posted on the course web page. The input and output spaces are continuous (i.e.,  $x \in \mathbb{R}^d$  and  $y \in \mathbb{R}$ ).

- (a) **[15 pts]** Regularized generalized linear regression: perform least square regression with the penalty term  $w^T w$ . Use monomial basis functions up to degree  $d$ :  $\{\prod_i (x_i)^{n_i} \mid \sum_i n_i \leq d\}$
- (b) **[15 pts]** Bayesian generalized linear regression: use monomial basis function up to degree  $d$  as described above. Assume the output noise is Gaussian with variance = 1. Start with a Gaussian prior over the weights  $\Pr(w) = N(0, I)$  with 0 mean and identity covariance matrix.
- (c) **[20 pts]** Gaussian process regression: assume the output noise is Gaussian with variance = 1. Use the following kernels:

- Identity:  $k(x, x') = x^T x'$
- Gaussian:  $k(x, x') = e^{-||x-x'||^2/2\sigma^2}$
- Polynomial:  $k(x, x') = (x^T x' + 1)^d$  where  $d$  is the degree of the polynomial

**What to hand in:**

- Your code for each algorithm.
- Regularized generalized linear regression:
  - Graph that shows the mean squared error based on 10-fold cross validation for degrees 1, 2, 3 and 4 of the monomial basis functions.
  - A discussion of the results and how the running time varies with the degree of the monomial basis functions.
- Bayesian generalized linear regression:
  - Graph that shows the mean squared error based on 10-fold cross validation for degrees 1, 2, 3 and 4 of the monomial basis functions.
  - A discussion of the results and how the running time varies with the degree of the monomial basis functions.
  - A discussion of the similarities and differences between regularized generalized linear regression and Bayesian generalized linear regression.
- Gaussian process regression:
  - The mean squared error based on 10-fold cross validation for the identity kernel.
  - Graph that shows the mean squared error based on 10-fold cross validation for the Gaussian kernel when we vary  $\sigma$  from 1 to 6 in increments of 1.
  - Graph that shows the mean squared error based on 10-fold cross validation for degrees 1, 2, 3 and 4 of the polynomial kernel.
  - A discussion of the results and how the running time varies.