Assignment 3: Non-linear regression

CS480/680 – Spring 2019

Out: June 12, 2019
Due: June 28 (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 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. **[80 pts]** Non-linear regression techniques.

   Implement the following regression algorithms. For a), b) and c), do not use any machine learning library, but feel free to use libraries for linear algebra and feel free to verify your results with existing machine learning libraries. For d) feel free to use a machine learning package such as Keras, TensorFlow or PyTorch to implement your neural network. Use the dataset posted on the course web page. The input and output spaces are continuous (i.e., $x \in \Re^d$ and $y \in \Re$).

   (a) **[20 pts]** Regularized generalized linear regression: perform least square regression with the penalty term $0.5w^Tw$. Use monomial basis functions up to degree d: $\{\prod_i (x_i)^{n_i} | \sum_i n_i \leq d\}$. A monomial of degree less than or equal to d is a product of variables (e.g., $\prod_i (x_i)^{n_i}$) where the sum of their exponents is less than or equal to d (e.g., $\sum_i n_i \leq d$).

   (b) **[20 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

   (d) **[20 pts]** Neural network: minimize the squared loss of a two-layer neural network with a sigmoid activation function for the hidden nodes and the identity function for the output node.

   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.
  - The best degree found by 10-fold cross validation and the squared error for the test set.
  - How does the running time vary 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.
  - The best degree found by 10-fold cross validation and the squared error for the test set.
  - How does the running time vary with the degree of the monomial basis functions?
  - What are the similarities and differences between regularized generalized linear regression and Bayesian generalized linear regression.
- Gaussian process regression:
  - The mean squared error of the test set for the identity kernel.
  - Graph that shows the mean squared error based on 10-fold cross validation for the Gaussian kernel when we vary σ from 1 to 6 in increments of 1. The mean squared error of the test set for the best σ.
  - Graph that shows the mean squared error based on 10-fold cross validation for degrees 1, 2, 3 and 4 of the polynomial kernel. The mean squared error of the test set for the best polynomial degree.
  - How does the running time vary?
- Neural network:
  - Graph that shows the mean squared error based on 10-fold cross validation as we vary the number of hidden units from 1 to 10 (in increments of 1).
  - The best number of hidden units found by 10-fold cross validation and the squared error for the test set.
  - How does the running time vary with the number of hidden units?