Compsci 571 HW4

Yilin Gao (yg95)

February 26, 2018

1 Constructing Kernels

2 Reproducing Kernel Hilbert Spaces

3 Convexity and KKT Conditions

(a) The Lagrangian function for the primal form is:

$$\min L(\mathbf{w}, \eta, \eta^*, \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\eta_i + \eta_i^*) + \sum_{i=1}^{n} a_i[y_i - \mathbf{w}^T\mathbf{x}_i - \epsilon - \eta_i]$$

$$+\sum_{i=1}^{n} b_i[\mathbf{w}^T\mathbf{x}_i - y_i - \epsilon - \eta_i^*] - \sum_{i=1}^{n} c_i\eta_i - \sum_{i=1}^{n} d_i\eta_i^*$$

It's KKT conditions are:

- Primal feasibility:

$$\begin{gathered} y_i - \mathbf{w}^T\mathbf{x}_i - \epsilon - \eta_i \le 0, i = 1, \dots, n \\ \mathbf{w}^T\mathbf{x}_i - y_i - \epsilon - \eta_i^* \le 0, i = 1, \dots, n \\ \eta_i \ge 0, i = 1, \dots, n \\ \eta_i^* \ge 0, i = 1, \dots, n \end{gathered}$$

- Dual feasibility:

$$\begin{gathered} a_i \ge 0, i = 1, \dots, n \\ b_i \ge 0, i = 1, \dots, n \\ c_i \ge 0, i = 1, \dots, n \\ d_i \ge 0, i = 1, \dots, n \end{gathered}$$

- Complementary slackness:

$$\begin{gathered} a_i[y_i - \mathbf{w}^T\mathbf{x}_i - \epsilon - \eta_i] = 0, i = 1, \dots, n \\ b_i[\mathbf{w}^T\mathbf{x}_i - y_i - \epsilon - \eta_i^*] = 0, i = 1, \dots, n \\ c_i\eta_i = 0, i = 1, \dots, n \\ d_i\eta_i^* = 0, i = 1, \dots, n \end{gathered}$$

- Lagrangian stationary:

$$\nabla_{\mathbf{w}L} = \mathbf{w} - \sum_{i=1}^{n}(a_i - b_i)\mathbf{x}_i = 0$$
$$\nabla_{\eta L} = C - \mathbf{a} - \mathbf{c} = 0$$
$$\nabla_{\eta^* L} = C - \mathbf{b} - \mathbf{d} = 0$$

With these conditions, we can transform Lagrangian function into dual form:

$$\max L(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{n}(a_i - b_i)y_i - \epsilon \sum_{i=1}^{n}(a_i + b_i) - \frac{1}{2}\sum_{i,j=1}^{n}(a_i - b_i)(a_j - b_j)\mathbf{x}_i^T \mathbf{x}_j$$

subject to

$$0 \le a_i, b_i \le C, i = 1, \ldots, n$$

(b) Support vectors are the points i such that $|y_i - \mathbf{w}^T \mathbf{x}_i| \le \epsilon$.

(c) Increasing ϵ makes the model less likely to overfit. Because the model penalizes the points that have training error larger than ϵ. If ϵ increases, the allowed/unpenalized training error increases, and the model tends to overfit less.

(d) Increasing C makes the model more likely to overfit. C is the penalty for each point that has training error larger than ϵ. If the penalty increases, the model will try to make points have smaller training error, and thus overfits.

(e) Assume we've computed the optimal dual variables as $\mathbf{a}^*$ and $\mathbf{b}^*$.

From one of the KKT conditions, we can get the optimal primal variable is $\mathbf{w}^* = \sum_{i=1}^{n}(a_i^* - b_i^*)\mathbf{x}_i$.

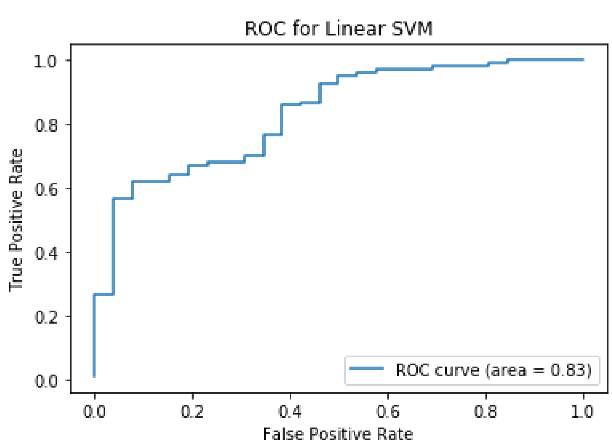
So for a new point $\mathbf{x}^{new}$, its evaluation is $f(\mathbf{x}^{new}) = \sum_{j=1}^{p} w_j^* x_j^{new} = \sum_{j=1}^{p}\sum_{i=1}^{n}(a_i^* - b_i^*)x_{ij}x_j^{new} = \sum_{i=1}^{n}(a_i^* - b_i^*)\mathbf{x}_i \cdot \mathbf{x}^{new}$.

4 SVM Implementation

(a) See `smv_classifier.py`.

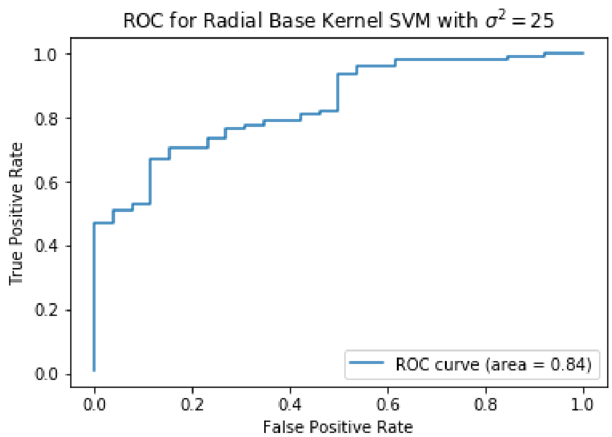
(b) **Note: for questions (b) and (c), I use `sklearn.mode_selection.train_test_split` to split the training and testing set with 2018 as the random seed. And I've noticed if I use `numpy` to generate indices with 2018 as the random seed and then split, the split is different.** Code for these 2 questions are in `q4.ipynb`.

The accuracy of the classifier on testing data is 0.8636363636363636. The ROC curve is like following:



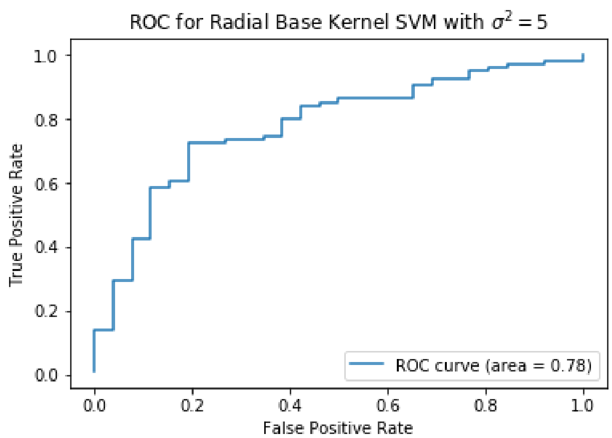
The AUC on testing data is 0.8316400580551523.

(c) For $\sigma^2 = 25$, the accuracy of the classifier on testing data is 0.8484848484848485. The ROC curve is like:



The AUC on testing data is 0.8388969521044993.

For $\sigma^2 = 5$, the accuracy for the classifier on testing data is 0.7954545454545454. The ROC curve is like:



The AUC on testing data is 0.7790275761973875.

The comparison between 2 σ^2 values suggests that for Gaussian kernel if we set σ^2 too small we may overfit.