Yeo Ying Xuan
1003835

50.007 Machine Learning
Homework 2

1 Support Vector Machines

1.1

(a) Determine the Kernel $K(x, y)$

$$x = [x_1 \quad x_2]^T$$

$$\varphi(x) = [1 \quad x_1^2 \quad \sqrt{2}x_1x_2 \quad x_2^2 \quad \sqrt{2}x_1 \quad \sqrt{2}x_2]^T$$

$$\varphi(y) = [1 \quad y_1^2 \quad \sqrt{2}y_1y_2 \quad y_2^2 \quad \sqrt{2}y_1 \quad \sqrt{2}y_2]^T$$

$$\begin{aligned} K(x,y) &= \varphi(x) \cdot \varphi(y) \\ &= 1 + x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 + 2x_1y_1 + 2x_2y_2 \\ &= (x_1y_1 + x_2y_2 + 1)^2 \\ &= (x^Ty + 1)^2 \end{aligned}$$

(b) Calculate the value of the Kernel $K(x, y)$

$$x = [1 \quad 2]^T$$

$$y = [3 \quad 4]^T$$

$$\begin{aligned} K(x,y) &= 1 + 1^2 \cdot 3^2 + 2(1 \cdot 2 \cdot 3 \cdot 4) + 2^2 \cdot 4^2 + 2(1 \cdot 3) + 2(2 \cdot 4) \\ &= 144 \end{aligned}$$

1.2

(1) Derive formulation of dual problem with soft margin

Primal Problem

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}w^Tw + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & d_i(w^Tx_i + b) - 1 + \xi_i \geq 0, \qquad i = 1, 2, \dots, N \\ & \xi_i \geq 0, \qquad i = 1, 2, \dots, N \end{aligned}$$

Write the constraints in standard form

$$1 - d_i(w^Tx_i + b) - \xi_i \leq 0$$

$$-\xi_i \leq 0$$

Lagrange Function

$$\begin{aligned} L(w, b, \xi, \alpha, \beta) &= \frac{1}{2}w^Tw + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i[1 - d_i(w^Tx_i + b) - \xi_i] + \sum_{i=1}^{N}\beta_i(-\xi_i) \\ &= \frac{1}{2}w^Tw + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\alpha_i\xi_i - \sum_{i=1}^{N}\alpha_id_iw^Tx_i - \sum_{i=1}^{N}\alpha_id_ib - \sum_{i=1}^{N}\beta_i\xi_i \end{aligned}$$

Satisfying KKT Conditions

- KKT condition 1: Partial derivative with respect to w parameter

$$\frac{\partial L(w,b,\xi,\alpha,\beta)}{\partial w} = 0$$

$$\frac{\partial L(w,b,\xi,\alpha,\beta\ )}{\partial w} = w - \sum_{i=1}^{N} \alpha_i d_i x_i$$

$$\Rightarrow w = \sum_{i=1}^{N} \alpha_i d_i x_i \qquad (1)$$

- KKT condition 2: Partial derivative with respect to b parameter

$$\frac{\partial L(w,b,\xi,\alpha,\beta\ )}{\partial b} = 0$$

$$\frac{\partial L(w,b,\xi,\alpha,\beta)}{\partial b} = -\sum_{i=1}^{N} \alpha_i d_i$$

$$\Rightarrow \sum_{i=1}^{N} \alpha_i d_i = 0 \qquad (2)$$

- KKT condition 3: Partial derivative with respect to ξ parameter

$$\frac{\partial L(w,b,\xi,\alpha,\beta)}{\partial \xi} = 0$$

$$C - \alpha_i - \beta_i = 0$$

$$\Rightarrow C = \alpha_i + \beta_i \qquad (3)$$

- KKT condition 4:

$$\alpha_{o,i}[d_i(w_o^T x_i + b_o) - 1 + \xi_i] = 0, \qquad i = 1, 2, \dots, N$$

- KKT condition 5: Set dual variables constraints

$$\alpha_i \geq 0$$

$$\beta_i \geq 0$$

$$i = 1, 2, \dots, N$$

Sub (1), (2) & (3) into $L(w,b,\xi,\alpha,\beta)$ to reduce to only α_i unknowns:

$$\because \sum_{i=1}^{N} \alpha_i d_i = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i d_i b = b \sum_{i=1}^{N} \alpha_i d_i = 0 \quad (b \text{ is a constant})$$

$$L(w,b,\xi,\alpha,\beta\ ) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i + \sum_{i=1}^{N} \alpha_i [1 - d_i(w^T x_i + b) - \xi_i] + \sum_{i=1}^{N} \beta_i(-\xi_i)$$

$$= \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i + \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \alpha_i d_i (w^T x_i) - \sum_{i=1}^{N} \alpha_i d_i b - \sum_{i=1}^{N} \alpha_i \xi_i - \sum_{i=1}^{N} \beta_i \xi_i$$

$$= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j x_i^T x_j + (\alpha_i + \beta_i)\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j x_i^T x_j - \sum_{i=1}^{N}\alpha_i\xi_i - \sum_{i=1}^{N}\beta_i\xi_i$$

$$= \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j x_i^T x_j$$

$$= \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j (\overrightarrow{x_i} \cdot \overrightarrow{x_j})$$

$$= Q(\alpha)$$

Deriving constrains of dual problem

- Since α_i & β_i are dual variables where $\alpha_i \geq 0$ & $\beta_i \geq 0$ (KKT condition 5)

$$\alpha_i = C - \beta_i \text{ (derived from KKT condition 3)}$$
$$\Rightarrow 0 \leq \alpha_i \leq C$$
$$\because C = \frac{1}{\lambda},$$
$$0 \leq \alpha_i \leq \frac{1}{\lambda}$$

- From KKT condition 2 derived earlier,

$$\sum_{i=1}^{N}\alpha_i d_i = 0$$

Dual Problem Formulation

$$max\ Q(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j x_i^T x_j$$
$$\text{subject to } 0 \leq \alpha_i \leq \frac{1}{\lambda}, \qquad i = 1, 2, \dots, N$$
$$\sum_{i=1}^{N}\alpha_i d_i = 0, \qquad i = 1, 2, \dots, N$$

Optimal Solution

For $0 < \alpha_i < C$,
$$d_i(w^T x_i + b) = 1$$

Multiply both sides by d_i,
$$d_i^2(w^T x_i + b) = d_i$$

$$\because d_i = \{-1, +1\},$$
$$d_i^2 = 1$$
$$w^T x_i + b = d_i$$

For support vector x_i,
$$b_o = d_i - w_o^T x_i$$

(2) Explain in which cases we would prefer to use soft margin rather than hard margin.

The performance of SVM should be evaluated based on the overall number of misclassifications for both training and testing data on both soft margin and hard margin SVM. When the overall number of misclassified points combined from training and test data is smaller with soft margin relative to hard margin, in such cases, we would prefer soft margin over hard margin.

In general, using soft margin can reduce the risk of overfitting of the training data by allowing some points to violate the margin via the introduction of slack variables. Having a large margin effectively corresponds to regularizing the weights of the support vectors, and as a result reduces the risk of overfitting. Hence, we tend to prefer a large margin (through maximization) because it enables us to generalize our model such that there is a higher possibility of performing better on test data.

As for a hard margin SVM, because of the fact that data has to be completely linearly separable, a single outlier or mislabeled point can potentially affect the resulting decision boundary drastically, which makes the SVM classifier overly sensitive to noise in the dataset. Using the same set of training data (in presence of outlier or mislabeled points), the margin for hard margin SVM can be much smaller than that of soft margin SVM.

1.3 SVM Kernel Implementation

- Refer to `HW2_1003835.ipynb` for code.

(a) Linear Kernel

```
Accuracy = 74.6032% (47/63) (classification)
```

(b) Polynomial Kernel

```
Accuracy = 53.9683% (34/63) (classification)
```

(c) Radial Basis Function (RBF) Kernel

```
Accuracy = 84.127% (53/63) (classification)
```

(d) Sigmoid Kernel

```
Accuracy = 79.3651% (50/63) (classification)
```

Radial Basis Function (RBF) kernel is chosen as it has the highest accuracy among the four different kernels tested on.

RBF kernels project vectors into an infinite dimensional space, by taking an infinite sum over polynomial kernels (Taylor expansion of exponential function). It is a stationary kernel such that it is invariant to translation: $K(x, y) = K(x + c, y + c)$, whereas linear kernel does not have this stationary property. The RBF kernel is also isotropic, where same amount of scaling occurs in all directions.

2 Logistic Regression

2.1

Context

- Binary classification where $y = 0$ or $y = 1$
- $h_\theta(x) = 0.35$

(1) False – Our estimate for $P(y = 0|x; \theta)$ is 0.35

$h_\theta(x) = P(y = 1|x; \theta) = 0.35$
Prediction $h_\theta(x)$ gives the probability of obtaining the value of $y = 1$, not $y = 0$.

(2) True – Our estimate for $P(y = 0|x; \theta)$ is 0.65

Given that this is a binary classification where it is either $y = 0$ or $y = 1$,
$P(y = 0|x; \theta) + P(y = 1|x; \theta) = 1$

Since $h_\theta(x) = P(y = 1|x; \theta)$,
$P(y = 0|x; \theta) = 1 - h_\theta(x) = 1 - 0.45 = 0.65$

(3) True – Our estimate for $P(y = 1|x; \theta)$ is 0.35

$h_\theta(x) = P(y = 1|x; \theta) = 0.35$
Prediction $h_\theta(x)$ gives the probability of obtaining the value of $y = 1$

(4) False – Our estimate for $P(y = 1|x; \theta)$ is 0.65

$h_\theta(x) = P(y = 1|x; \theta) = 0.35$
Prediction $h_\theta(x)$ gives the probability of obtaining the value of $y = 1$

2.2 Formulate decision boundary of classifier

$$h_\theta(x) = 6 - x_1$$

$$Given\ threshold = 0.5,$$
$$y = 1\ if\ x_1 \leq 6$$
$$y = 0\ if\ x_1 > 6$$

2.3 Formulate decision boundary of classifier

$$h_\theta(x) = -9 + x_1^2 + x_2^2$$

$$Given\ threshold = 0.5,$$
$$y = 1\ if\ x_1^2 + x_2^2 \geq 9$$
$$y = 0\ if\ x_1^2 + x_2^2 < 9$$

2.4 Benefit of optimizing log-likelihood rather than likelihood

Looking at the maximum likelihood equation, when each data point is independent of one another, the total probability of observing all of the data points $x^{(i)}, i = 1,2, \dots, n$ (ie. joint probability distribution) is the product of observing each data point individually. Having in mind that probability values range between 0 and 1 (inclusive), when n is large (ie. a large dataset), multiplying many small probabilities together can be numerically unstable, hence prone to numerical underflow.

Taking the natural logarithm of the maximum likelihood function will transform the equivalent optimization problem from product into summation term, based on the property $\ln(ab) = \ln(a) + \ln(b)$. Since the natural logarithm is a monotonically increasing function, the optima points are preserved, hence we can work with the log-likelihood instead of the original likelihood function for simpler calculation. The summation form is both more numerically stable and easier to differentiate than the original maximum likelihood formulation.

It is of common practice to phrase the optimization problem as a minimization of the cost function. Maximum likelihood is hence formulated as the negative log-likelihood, as shown in Equation 2.