Optimización Numérica: Laboratorio 3

rodrigo.mendoza@itam.mx

24 Septiembre 2019

1 Programación Cuadrática

Supongamos que tenemos un conjunto de datos $\mathcal{D} = \{(\mathbf{x}_i, y_i) \in \mathbb{R}^p \times \{-1, 1\} : i \in [m]\}$. Se dice que los puntos son *linealmente separables* si existe un hiperplano $\mathbf{w}^T\mathbf{x} + b = 0$ tal que $\mathbf{w}^T\mathbf{x}_i + b > 0$ si $y_i > 0$ y $\mathbf{w}^T\mathbf{x}_i + b < 0$ si $y_i < 0$ $\forall i \in [m]$. Queremos encontrar el hiperplano $(\mathbf{w}, b)$ que mejor separe los datos. Una *máquinas de soporte vectorial* encuentra el hiperplano que maximiza el *margen*, es decir, la distancia mínima entre el hiperplano y un punto en $\mathcal{D}$.

1. Demuestre que $\min_{\mathbf{v}}\{\|\mathbf{x} - \mathbf{v}\| : \mathbf{w}^T\mathbf{v} + b = 0\} = |\mathbf{w}^T\mathbf{x} + b|$
   Pista: Demuestre que $\mathbf{v} = \mathbf{x} - (\mathbf{w}^T\mathbf{x} + b)\mathbf{w}$ está en el hiperplano y que $\|\mathbf{x} - \mathbf{u}\|^2 \geq \|\mathbf{x} - \mathbf{v}\|^2$ para todo $\mathbf{u}$ en el hiperplano.

2. El modelo *duro* de *máquinas de soporte vectorial* puede formularse como:

$$\underset{(\mathbf{w},b):\|\mathbf{w}\|=1}{\arg\max} \min_{i\in[m]} |\mathbf{w}^T\mathbf{x}_i + b| \quad \text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) > 0 \quad \forall i \in [m]. \tag{1}$$

   Demuestre que este modelo puede reescribirse como:

$$\underset{(\mathbf{w},b):\|\mathbf{w}\|=1}{\arg\max} \min_{i\in[m]} y_i(\mathbf{w}^T\mathbf{x}_i + b). \tag{2}$$

3. Use este resultado para demostrar que el puede resolverse con

$$\begin{aligned} (\mathbf{w}_0, b_0) &= \underset{(\mathbf{w},b)}{\arg\max} \|\mathbf{w}\|^2 \quad \text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i \in [m] \\ \hat{\mathbf{w}} &= \mathbf{w}_0/\|\mathbf{w}_0\| \\ \hat{\mathbf{b}} &= \mathbf{b}_0/\|\mathbf{b}_0\| \end{aligned} \tag{3}$$

   Pista: Suponga que $(\mathbf{x}^*, b^*)$ es una solución de (2) y considere $\gamma^* = \min_{i\in[m]} y_i((\mathbf{w}^*)^T\mathbf{x}_i + b^*)$. Demuestre que $\mathbf{w}_0 \leq 1/\gamma^*$ y use esto para demostrar que $(\hat{\mathbf{w}}, \hat{b})$ es óptimo.

4. Un *programa cuadrático* tiene la forma:

$$\begin{array}{ll} \min_{\mathbf{x}} & \frac{1}{2}\mathbf{x}^T\mathbf{P}\mathbf{x} + \mathbf{q}^T\mathbf{x} \\ \text{s.t.} & \mathbf{G}\mathbf{x} \leq \mathbf{h} \\ & \mathbf{A}\mathbf{x} = \mathbf{b} \end{array} \tag{4}$$

5. Reescriba (3) como un programa cuadrático.

6. Implemente la función `hard_svm` en Python que resuelva (1) usando el método `cvxopt.solvers.qp` (https://cvxopt.org/userguide/coneprog.html#quadratic-programming) [1].

```python
def hard_svm(X, y):
    '''(X,y):  Datos de entrenamiento [X.shape=(m,p), y.shape=(m,)]
               X,y matrices de numpy
       (w,b):  Hiperplano [w.shape=(p,), b.shape=(1,)]
    '''
    ...
    return w, b
```

[1] Si lo necesita, puede encontrar un tutorial aquí https://courses.csail.mit.edu/6.867/wiki/images/a/a7/Qp-cvxopt.pdf

7. Implmente la siguiente función para generar un conjunto de datos linealmente separables:

```
import numpy as np
def datos(N):
    np.random.seed(1111)
    n = int(N/2)
    cov = [[1, 0], [0, 1]]
    X1 = np.random.multivariate_normal([-9,10], cov, n)
    X2 = np.random.multivariate_normal([7,-8], cov, n)
    X = np.concatenate((X1, X2), axis=0)
    y = np.concatenate((np.ones(n), -1*np.ones(n)), axis=0)
    return X, y.squeeze()
```

8. Pruebe su función `hard_svm` con `(X,y) = datos(200)`. Grafique los puntos con el hiperplano que encontró.