Optimizació Numérica

Manuel Sarmiento Navarro

7 de octubre de 2019

1. Laboratorio 2

Laboratorio 2

2. Laboratorio 3

1. Demostrar que $\min_v\{||x-v|| : w^T v + b = 0\} = |w^T x + b|$
2. Demostrar que el modelo *duro* de *máquinas de soporte vectorial*:

$$\begin{gathered}\underset{(w,b):||w||=1}{\arg\max}\ \min_{i\in[m]} |w^T x_i + b| \\ \text{s. a. } y_i(w^T x_i + b) > 0 \quad \forall \quad i \in [m]\end{gathered}$$

   se puede reescribir como

$$\underset{(w,b):||w||=1}{\arg\max}\ \min_{i\in[m]} y_i(w^T x_i + b)$$

3. Probar que este problema se puede resolver con

$$\begin{gathered}(w_0, b_0) = \underset{(w,b)}{\arg\max}\, ||w||^2 \\ \text{s. a. } y_i(w^T x_i + b) \geq 1 \quad \forall \quad i \in [m] \\ \hat{w} = \frac{w_0}{||w_0||}, \quad \hat{b} = \frac{b_0}{||w_0||}\end{gathered}$$

4. Reescribir el problema anterior como un programa cuadrático de la forma

$$\begin{gathered}\frac{1}{2}x^T P x + q^T x \\ \text{s. a. } Gx \leq h \\ Ax = b\end{gathered}$$

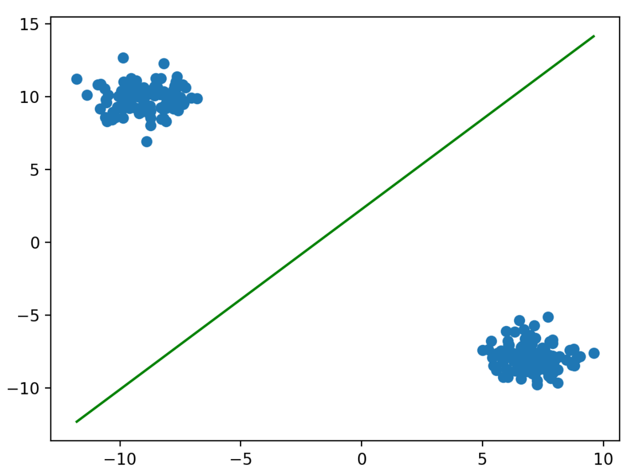
5. Escribir la función `hard_svm()` en Python que resuelva el problema anterior usandlo `cvxopt`

```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,)]
  '''
  P = matrix(np.concatenate((np.concatenate((np.identity(X.shape[1], float), np.zeros((X.shape[1],1),float)), axis=1), np.zeros((1,(X.shape[1]+1)),float)), axis=0), tc='d')
  q = matrix(np.zeros((X.shape[1]+1,1),float), tc='d')
  y.shape = (X.shape[0],1)
  X = np.concatenate((X, np.ones((X.shape[0],1))), axis=1)
  G = -y * X
  G = matrix(G, tc='d')
  h = matrix(-np.ones(X.shape[0]), tc='d')

  sol = solvers.qp(P,q,G,h)
  print(sol['x'])
  w = np.array(sol['x'])
  w, b = w[0:-1,:], w[-1,:]

  return w, b
```

6. Al graficar la el hiperespacio que resulta de aplicar la función `hard_svm()` a los puntos `(X,y) = datos(200)` se obtiene:



3. Laboratorio 4

Laboratorio 4