Tarea 3: SVM y Random Forests (Parte teórica)

Entrega: Jueves 03 de octubre, 23:59 (Gradescope)

Profesor: Pablo Estévez V.
Auxiliar: Ignacio Reyes J.
Semestre: Primavera 2019

1. Pregunta 1

Considere el siguiente conjunto de entrenamiento:

$$C_1 : (-2,-2); (-1,1)\{+1\}$$
$$C_2 : (1,1); (2,-2)\{-1\}$$

La regla de decisión de un clasificador en su forma primal es $h(\vec{x}) = sgn(\langle w, \vec{x}_i \rangle + b)$. Encuentre los pesos w y el sesgo b del clasificador h_{max} de máximo margen. Para ello realice los siguientes pasos:

1. Grafique el conjunto de entrenamiento y determine visualmente cuáles de los ejemplos son vectores de soporte de h_{max}.

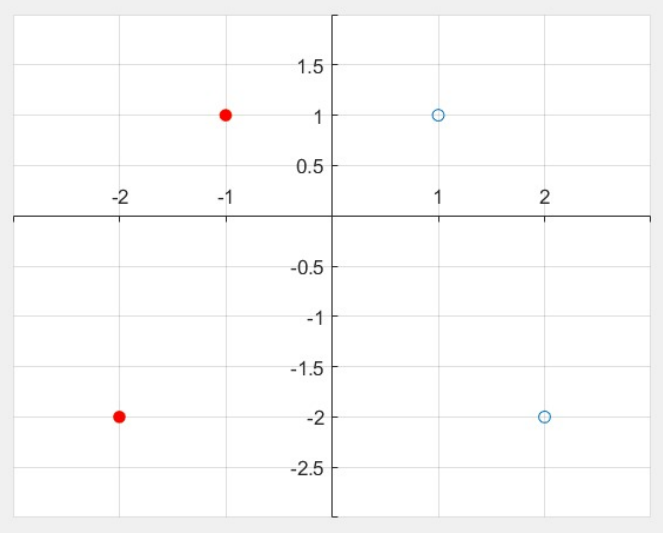

Figura 1: Grafico con el conjunto de entrenamiento

Sol: visualmente los puntos que originam a maior margem h max (-1, 1) e (1, 1), luego estos son los vectores de suporte

2. Formule el problema de optimización primal (maximización del margen sujeto a que los ejemplos estén bien clasificados).

   Sol: Los clasificadores SV están basados en la clase de hiperplanos

$$(w.x =) + b = 0$$

$$w \in R^d, b \in R$$

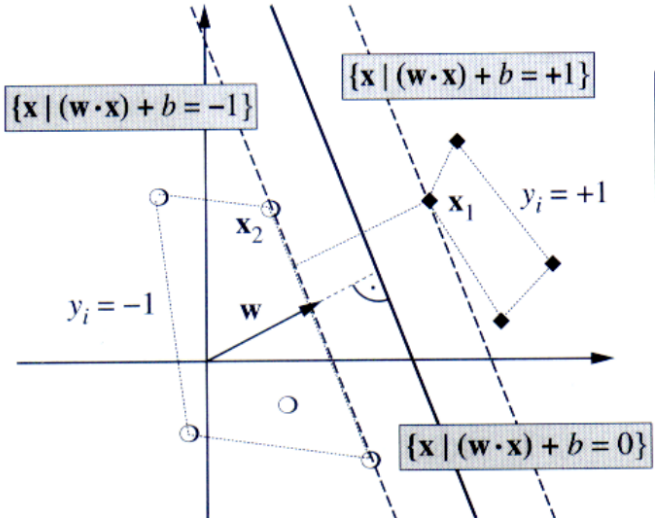

Figura 2: Hyperplano y ecuaciones en el plano de 2 dimensiones

$$(w \cdot x_1) + b = +1$$
$$(w \cdot x_2) + b = -1$$
$$=> (w \cdot (x_1 - x_2)) = 1 - b + 1 + b = 2$$
$$margen = (\frac{w}{||w||} \cdot (x_1 - x_2)) = \frac{2}{||w||}$$

Nuestra función objetivo es entonces maximizar la margen:

$$\text{MAX } margen = \frac{2}{||w||} \text{ (que es lo mismo que hacer)} \rightarrow \text{MIN } ||w|| \rightarrow \text{MIN } \frac{1}{2}||w||^2$$

Para encontrar el hiperplano óptimo, el problema de optimización primal es:

$$\text{MIN } margen = \frac{1}{2}||w||^2,$$
$$\text{S.A. } y_i[(w \cdot x_i) + b] \geq 1 \quad \text{i =1, ... , n; } y_i \in \{\pm 1\}$$

3. Dados los vectores de soporte, identifique qué restricciones se activan. Con esto obtendrá tantas ecuaciones como vectores de soporte hayan.

   Sol:

   El hiperplano separador debe satisfacer las siguientes restricciones:

   $$\begin{array}{ll}(w \cdot x_i) + b \geq +1 & \text{if } y_i = +1 \\ (w \cdot x_i) + b \leq -1 & \text{if } y_i = -1\end{array}$$

   Estas restricciones se pueden reducir a una única expresión

   $$y_i[(w \cdot x_i) + b] \geq 1 \quad \text{i} = 1, \ldots, \text{n}$$

   $$y_i[(w \cdot x_i) + b] - 1 = 0 \quad \text{i para puntos x en el hiperplano separador}$$

   Dado que los vectores en este hiperplano separador se consideran vectores de soporte, tenemos tantas restricciones activas como vectores de soporte.

4. Aplique las condiciones de Karush-Kuhn-Tucker (KKT) sobre el problema de optimización, tomando en cuenta que las restricciones activas son las únicas con multiplicadores no nulos.

   Sol:

   Este problema de optimización con restricciones se trata introduciendo multiplicadores de Lagrange $\alpha_i \geq 0$ y el Lagrangiano:

   $$L = \tfrac{1}{2}||w||^2 - \textstyle\sum_{i=1}^{n} \alpha_i y_i x_i) + b] - 1)$$

   Las condiciones de complementariedad de Karush-Kuhn-Tucker se logra con derivadas de L con respecto a las variables principales y estas tienen que hacerse cero:

   $$\begin{array}{l}\frac{\partial L(w,b,\alpha)}{\partial w} = 0 \\ \frac{\partial L(w,b,\alpha)}{\partial b} = 0\end{array}$$

   conduce a que,

   $$\begin{array}{c}\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i <=> w = \sum_{i=1}^{n} \alpha_i y_i x_i \\ \frac{\partial L(w,b,\alpha)}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0\end{array}$$

   Aplicando esses resultados a $L(w, b, \alpha)$:

   $$\begin{array}{c}L = \tfrac{1}{2}(\sum_{i=1}^{n} \alpha_i y_i x_i)(\sum_{j=1}^{n} \alpha_j y_j x_j) - \sum_{i=1}^{n} \alpha_i y_i x_i \cdot (\sum_{j=1}^{n} \alpha_j y_j x_j) - (\sum_{i=1}^{n} \alpha_i y_i b) - \sum_{i=1}^{n} \alpha_i \\ M(\alpha) = \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2}(\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i \cdot x_j)\end{array}$$

5. Encuentre los valores de todos los multiplicadores, de los pesos w y el sesgo b del clasificador h_{max} de máximo margen.

   Sol:

   Con la ayuda del software matlab, se calculó el alfa utilizando las herramientas disponibles para resolver mediante programación cuadrática.

```
H; %Matriz Hessiana

f=-ones(n,1);

A= -1.*[1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
Aeq=y;
b= [0;0;0;0];
beq= 0;

alpha = quadprog(H,f,A,b,Aeq,beq);
```

Figura 3: función utilizada y sus argumentos

```
alpha =

    0.5000
    0.0000
    0.0000
    0.5000
```

Resultando en $\alpha > 0$ para vectores de soporte y $\alpha = 0$ para el resto.

A partir de la solucíon del dual, α, aplicamos $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ para obtener w

Sean $S = \{i : \alpha > 0\}$ los indices de los vectores soporte. Por las condiciones de holgura complementaria, para cada $i \in S$,

$$b = \frac{1 - y_i w \cdot x_i}{y_i} = y_i - w \cdot x_i$$

```
pesos =

   -1.0000         0


sesgo =

    0
```

6. Grafique la frontera de decisión del clasificador h_{max} de máximo margen.

   Sol: Tomando los valores de los pesos y el bias y las ecuaciones del hiperplano óptimo, tendremos el siguiente gráfico con el límite de decisión:

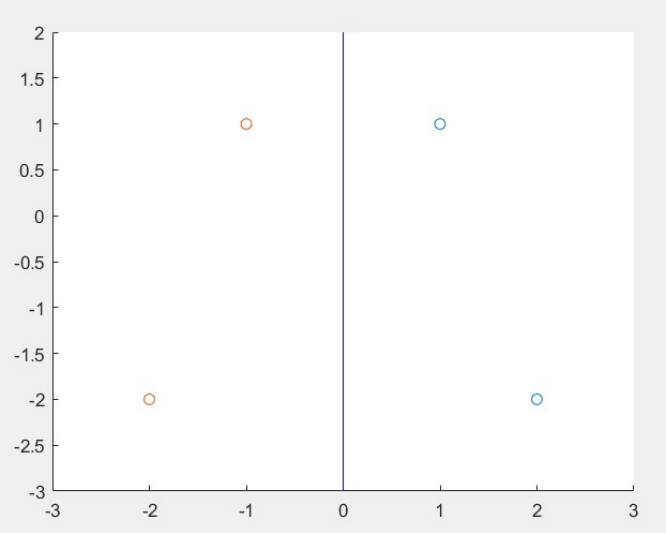

Figura 4: Grafico con el conjunto de entrenamiento y su frontera de decisión óptima

2. Pregunta 2

1. Imagine que está entrenando un modelo Random Forests. Dada una característica elegida para dividir el espacio ¿qué criterio se utiliza para colocar el umbral de clasificación? ¿Cómo se decide qué característica se utilizará para dividir el conjunto de muestras en un nodo?.

   Sol: En los métodos basados en árboles, el espacio se segmenta en varias regiones simples mediante reglas de partición. El espacio de entrada se divide en J regiones distintas disjuntas $R_1, R_2, ..., R_J$.

   Las regiones son cajas (cajas), son rectángulos en alto dimensión.

   El objetivo es encontrar cajas $R_1, R_2, ..., R_J$ que minimicen la suma de los residuos cuadráticos (RSS: residual sum squares):

   $$\sum_{j=1}^{J} \sum_{i \in R_J} (y_i - \hat{y}_{R_J})^2$$

   Donde $\hat{y}_{R_J}$ es el promedio de las observaciones en la región J.

   Para se decidir qué característica se utilizará para dividir el conjunto se selecciona la variable de predicción X_j y el punto de corte s que producen las regiones (por ejemplo $R_{izquierda}$ y $R_{derecha}$) (medios planos):

   $$R_{izquierda}(j, s) = \{X | X_j < s\} \text{ y } R_{derecha}(j, s) = \{X | X_j \geq s\}$$

   Una vez que se ha hecho una partición, el proceso se repite para particionar una de las regiones previamente identificadas y se repite hasta alcanzar un criterio de detención.

2. En los modelos Random Forests cada árbol de decisión clasifica utilizando un subconjunto de las características disponibles. Explique por qué se hace esto.

   Sol: Los modelos de Random Forests son metodos iterativos que comienzan un conjunto de datos y los subdivide en orden. Si hacemos sobre todo lo conjunto de datos, puede conducir a sobreajuste si los parametros crean arboles muy complejos y se torna en un modelo con alta varianza y bajo sesgo. El Bootstrap o Bootstrap aggregating(Bagging) es un metodo para combatir este problema creando varios subconjuntos, creando arboles para esos subconjuntos y presentando una media y varianza de este subconjunto. Estos son métodos conocidos como ensamble learning, si promedia los resultados y, por lo tanto, disminuye la varianza y el error.

3. ¿Qué son los métodos de kernel? ¿Qué ventaja tiene llevar los puntos a un espacio distinto? ¿Es necesario conocer la función que mapea los puntos al nuevo espacio para aplicar los métodos de kernel?

   Sol: El término kernel proviene del uso de tipos de funciones en el campo de operadores integrales estudiados por Hilbert y otros. Una función k que da origen a un operador T_k via $(T_k f)(x) = \int k(x, y) f(y) dy$ se llama el kernel de T_k . Una forma más sencilla de interpretar los

métodos de kernel es interpretar que pueden ser vistos como productos puntos generalizados. De hecho cualquier producto punto es un kernel.

Muchas veces es necesario hacer una extensión a reglas no lineales. Es posible que una regla de clasificacíon lineal no sea apropiada paralos datos originales $x_1, ..., x_n$ pero sí para los datos transformados$\Phi(x_1), ..., \Phi(x_n)$, donde $\Phi : R^d \rightarrow H$ para un espacio de Hilbert H que tiene mayor dimensión.

Para construir máquinas SV, el algoritmo del hiperplano óptimo debe ser aumentado mediante la computación de productos puntos en espacios de características no- linealmente relacionados con el espacio de entradas. La idea básica es mapear los datos en un espacio de productos puntos d.

Nuestro problema de optimización dual se convierte en:

$$M(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}(\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \Phi(x_i) \cdot \Phi(x_j))$$

En la practica, puede ser difícil calcular los productos escalares $\langle \Phi(x_i), \Phi(x_j) \rangle_H$, pero no es necesario conocer la función que mapea los puntos al nuevo espacio ya que tenemos un punto de producto en la función y solo requiere de la evaluación de productos puntos (esto se llama el truco del kernel)

$$k(x, y) = \Phi(x) \cdot \Phi(y).$$