Taller de Máquinas de Soporte Vectorial (SVM): Clasificación y Regresión

Juan Felipe Pérez, Julián D. Arias Londoño
Departamento de Ingeniería de Sistemas
Universidad de Antioquia, Medellín, Colombia
jdarias@udea.edu.co

23 de mayo de 2015

1. Marco teórico

Hasta este momento se han estudiado varias técnicas de modelamiento que permiten resolver problemas de clasificación. La utilización de estas técnicas genera una frontera de desición y permite diseñar una estrategia para clasificar nuevas muestras. Sin embargo, algunos modelos de clasificación pueden proporcionar diferentes fronteras de decisión, que de acuerdo con el error de entrenamiento, podrían ser equivalentes, es decir, producen el mismo nivel de error en el conjunto de muestras con las cuales se entrena un sistema. Surge entonces la pregunta de si es posible escoger una de las fronteras como la mejor, usando para ello las muestras de entrenamiento pero considerando como mejor, aquella frontera que pueda proporcionar un mejor resultado sobre muestras que el sistema aún no conoce. Una técnica cuyo objetivo es encontrar dicha frontera es llamada Máquina de Soporte Vectorial o SVM por sus siglas en inglés.

1.1. Clasificasión

Las Máquinas de Soporte Vectorial son un tipo de modelos de aprendizaje que buscan encontrar la mejor frontera de decisión, la cual se encuentra en un punto medio entre las dos clases a clasificar, utilizando como criterio de ajuste la máximización del margen, entendiendo margen como la distancia más corta entre la frontera de decisión y cualquiera de las muestras [1].

Para realizar el entrenamiento del modelo se usa como parámetros X y t, donde el primero son la muestras de entrenamiento y el ultimo las clases a clasificar, y $t \in -1, 1$. Si $t_n = -1$ para $y(x_n) < 0$ y $t_n = 1$ para $y(x_n) > 0$, entonces en una solución en la que todos los puntos se encuentran bien clasificados se cumple que $t_n y(x_n) > 0 \ \forall (x_n, t_n)$. Teniendo en cuenta que la distancia perpendicular de un punto $\mathbf{x}$ a un hiperplano definido por $y(x) = 0$ está dado por $|y(x)|/\|w\|$, lo que es equivalente a:

$$\frac{t_n y(x_n)}{\|w\|} = \frac{t_n(w^T x + b)}{\|w\|} \tag{1}$$

por lo tanto la solución de máximo margen se encuentra resolviendo:

$$\arg\max_{w,b} \left\{ \frac{1}{\|w\|} min_n \left[(t_n(w^T x_n + b \right] \right\} \tag{2}$$

Aunque la función anterior corresponde a un criterio de entrenamiento de maximización del margen, el proceso de optimización allí planteado es muy complejo. Por esa razón se debe encontrar una forma alternativa.

Teniendo en cuenta que si se reescala el vector **w** como kw y b como kb, siendo k un valor real, la distancia $t_n y(x_n)/\|w\|$ sigue sin cambiar, se puede asumir que para el punto más cercano a la superficie $t_n(w^T x + b) = 1$, por consiguiente todos los puntos cumplirán la condición:

$$t_n(w^T x + b) >= 1, n = 1, ..., N$$

Entonces el término a minimizar en la función criterio anterior es igual a 1, por lo tanto ahora sólo se requiere maximizar $1/\|w\|$ o lo que es equivalente a minimizar $\|w\|^2$, entonces:

$$\arg\max_{w,b} \frac{1}{2}\|w\|^2 \text{ Sujeto a } t_n(w^T x_n + b) >= 1 \tag{3}$$

Para solucionar el problema se introducen multiplicadores de Lagrange $a_n >= 0$, para cada restricción

$$\mathcal{L}(w, b, a) = \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N} a_n \left\{ t_n(w^T x_n + b) - 1 \right\} \tag{4}$$

Derivando con respecto a **w** y a b e igualando a cero se obtienen las siguientes expresiones:

$$w = \sum_{n=1}^{N} a_n t_n x_n$$

$$0 = \sum_{n=1}^{N} a_n t_n$$

Reemplazando las dos expresiones anteriores, en la función criterio original y usando la equivalencia $\frac{1}{2}\|w\|^2 = \frac{1}{2} w w^T$ y reescribiendo:

$$\widetilde{\mathcal{L}}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m x_n^T x_m \tag{5}$$

$$\text{Sujeto a } a_n >= 0, \sum_{n=1}^{N} a_n t_n = 0$$

Como se puede ver la función objetivo que se quiere optimizar no depende del vector de pesos **w** sino únicamente del producto $x_n^T x_m$. La gran fortaleza de las SVM radica en que el producto punto anterior se puede reemplazar por cualquier función $k(x_n, x_m)$ que cumpla las condiciones de Mercer.

Usando la formulación anterior y resolviendo el problema de optimización cuadrático, la función de decisión se convierte en:

$$y(x) = \sum_{n=1}^{N} a_n t_n k(x_n, x_m) + b \tag{6}$$

Y se satisfacen las siguientes condiciones:

$$a_n >= 0$$
$$t_n y(x_n) - 1 >= 0$$
$$a_n \{t_n y(x_n) - 1\} = 0$$

Los puntos con $a_n = 0$, no aparecen en la sumatoria y pueden descartarse, los demás se conocen como **vectores de soporte**. Estos vectores son los individuos más cercanos a la frontera, permitiendo utilizar solo estas muestras para realizar la clasificación, considerando que los vectores de soporte son suficientes para determinar la pertenencia a una clase o a otra, puesto que las muestras que están más lejanas a la línea de decisión son las que tienen menor incertidumbre y se consideran perfectamente clasificadas, sin aportar mucha información a la decisión tomada acerca de las nuevas muestras que se encuentren cerca de la frontera [1].

1.1.1. Clases no separables linealmente

La formulación del modelo SVM hecho hasta ahora asume que las clases son separables, al menos en un espacio de alta dimensión. Sin embargo, las distribuciones de las clases pueden estar traslapadas y por consiguiente es necesario modificar la SVM para que permita que algunas muestras queden mal clasificadas [1]. Las muestras que pertenecen a una clase y quedan al lado contrario de la frontera se consideran como mal clasificadas, por lo cual se hace uso de un margen, el cual permite que estas muestras sean clasificadas de forma correcta haciendo el cálculo de la menor distancia a la frontera y a los límites del margen. Esta margen es llamada *imagen de relajación*, porque le da al modelo un cierto grado de libertad. El modelo tiene un parámetro C el cual es el encargado de controlar el sobreajuste en términos de los vectores de soporte. Si $C \to \infty$, no hay valor que regularice el modelo, obteniendo el modelo original. La condición que se debe cumplir es:

$$0 <= a_n <= C$$

1.1.2. Estrategia *One vs All*

El modelo SVM esta diseñado para resolver problemas de clasificación biclase, es decir, el modelo solo funciona para problemas de no más de dos clases, por este motivo se hace

necesario usar alguna técnica que permita aplicar la SVM en un problema multiclase. Una de la técnicas utilizadas es la estrategia *One vs All*, la cual consiste en entrenar un clasificador por cada clase y haciendo al resto de las clases como si fueran de una sola clase, lo que hace el problema biclase. Esto se clase por cada clase, obteniendo un modelo diferente por cada clase que tenga el problema.

Al momento de hacer la predicción sobre una muestra nueva se debe evaluar esta muestra en cada uno de los clasificadores, obteniendo que la muestra pertenece a una de las clases, es decir un 1 en uno de los clasificadores y 0 en el resto. En el caso de las SVM el resultado es un 1 en un clasificador y -1 en el resto. Otro resultado que se puede obtener es que varios clasificadores devuelvan un 1 en la respuesta, esto significa que la muestra fue clasificada en varias clases. Para solucionar este problema se debe evaluar la muestra con los datos obtenidos en los modelos en conflicto, y dependiendo del modelo empleado evalúa y se determina a cual de las clases pertenece de forma diferente, en el caso de las máquinas de soporte vectorial se utiliza la ecuación 6, utilizando los vectores de soporte y la b de cada modelo, y la función en la cual se obtenga el mayor valor se considera que la muestra hace parte de esa clase.

Un ejemplo seria en un problema de tres clases clase 1, clase 2 y clase 3. Se entrena un clasificador para la clase 1, considerando a las clases 2 y 3 como si hicieran parte de la misma clase, poniendo un 1 de etiqueta en la clase 1, y un -1 a las clases 2 y 3, y se entrena el modelo. Este mismo procedimiento se repite con las otras dos clases. Al evaluar una muestra nueva x se obtiene un -1 con el clasificador para la primera clase y un 1 en los clasificadores de las otras dos clases, lo que significa que esta muestra se clasifico como si hiera parte de la clase 2 y de la clase 3. Lo siguiente es evaluar la muestra con la ecuación 6 y usando los vectores de soporte y la b obtenidos en el entrenamiento de cada modelo, de la siguiente manera:

$$y_2(x) = \sum_{n2=1}^{N} a_{n2} t_{n2} k(x_{n2}, x_m) + b_2$$

$$y_3(x) = \sum_{n3=1}^{N} a_{n3} t_{n3} k(x_{n3}, x_m) + b_3$$

Después de evaluar estas dos funciones se considera que la muestra x hace parte de la clase en el cual el valor y sea el menor, en el ejemplo si y_2 es menor que y_3 se concluye que la muestra x pertenece a la clase 2.

1.2. Regresión

Ahora expandamos las máquinas de soporte vectorial a problemas de regresión. En regresión lineal, se minimiza una función de error minimizada:

$$\frac{1}{2} \sum_{n=1}^{N} \{y_n - t_n\}^2 + \frac{\lambda}{2} \|w\| \tag{7}$$

En SVM esta función es reemplazada por una *función de error insensible a* ϵ, la cual adquiere un error igual a 0 si el valor absoluto de la diferencia entre $y(x)$ y el objetivo t es menor que ϵ, donde $\epsilon > 0$.

$$E_\epsilon(y(x) - t) = \begin{cases} 0, \text{ si } |y(x) - t| < \epsilon; \\ |y(x) - t| - \epsilon, \text{ en otro caso} \end{cases}$$

Se puede expresar el problema de optimización introduciendo variables de de *relajación*. Por cada punto x_n, se necesitan dos variables $\xi >= 0$ y $\widehat{\xi} >= 0$, donde $\xi > 0$ corresponde a un punto para el que $t_n > y(x_n) + \epsilon$ y $\widehat{\xi} > 0$ corresponde a un punto para el que $t_n < y(x_n) - \epsilon$.

La condición para que un punto este dentro de el margen de ϵ es que $y_n - \epsilon <= t_n <= y_n + \epsilon$, donde $y_n = y(x_n)$. Introduciendo las variables de relajación permite a ciertos puntos estar fuera del margen, estas condiciones son

$$t_n <= y(x_n) + \epsilon + \xi_n$$
$$t_n >= y(x_n) - \epsilon - \widehat{\xi}$$

La función de error esta dada por:

$$C \sum_{n=1}^{N} (\xi_n + \widehat{\xi_n}) + \frac{1}{2}\|w\|^2 \tag{8}$$

Esta función debe ser minimizada sujeta a las restricciones $\xi_n >= 0$ y $\widehat{\xi_n} >= 0$. Esto se puede lograr introduciendo los multiplicadores de Lagrange $a_n >= 0, \widehat{a_n} >= 0, \mu_n >= 0, \widehat{\mu_n} >= 0$ y se optimiza la función. Luego substituyendo por $y(n)$ usando $y(x) = w^T\phi(x) + b$ y obteniendo las derivadas con respecto a w, b, ξ_N y $\widehat{\xi_n}$ e igualando a cero se eliminan las variables correspondientes del Lagrangeciano, se obtiene:

$$\widetilde{\mathcal{L}}(a, \widehat{a}) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(a_n - \widehat{a}_n)(a_m - \widehat{a}_m)k(x_n, x_m) - \epsilon\sum_{n=1}^{N}(a_n + \widehat{a}_n) + \sum_{n=1}^{N}(a_n - \widehat{a}_n)t_n \tag{9}$$

Esta función se debe maximizar con respecto a a y $\widehat{a}_n$, donde se ha introducido el kernel $k(x, x') = \phi(x)^T\phi(x')$. De nuevo, esta es una maximización restringida, y para encontrarla se evidencia que $a_n >= 0$ y $\widehat{a}_n >= 0$ ambas restricciones son requeridas porque estos son multiplicadores de Lagrange. Otra vez se tienen las restricciones:

$$0 <= a_n <= C$$
$$0 <= \widehat{a}_n <= C$$
$$\text{Además de } \sum_{n=1}^{N}(a_n - \widehat{a}_n) = 0$$

Por ultimo remplazando y reacomodando se tiene que

$$y(x) = \sum_{n=1}^{N} (a_n - \widehat{a}_n) k(x, x_n) + b \tag{10}$$

Esta ecuación es expresada en términos de la función kernel [1].

2. Ejercicios

1. Adjunto a este taller encontrará los archivos: *Main.m, entrenarSVM.m, testSVM.m, evaluarFuncioSVM.m y kernelmat.m.* El archivo *Main.m* es el script principal, el cual lleva a cabo un experimento de clasificación y de regresión. Una vez corra el Script principal se le solicitará ingresar el numeral del punto que desea resolver (para regresión ingrese 1 ó ingrese 2 para clasificación). Analice con cuidado el script y comprenda como esta construido.

   Descargue la librería LS-SVMlab del link: `http://www.esat.kuleuven.be/sista/lssvmlab/`. Use los comandos *trainlssvm*, para entrenar el modelo, y *simlssvm*, para evaluarlo. La función *trainlssvm* retorna el modelo para evaluar la SVM y recibe como parámetro un modelo el cual debe tener: la matriz X con los datos de entrenamiento, el vector Y con los valores de las muestras, el tipo de modelo que se desea implementar, 'f' si es una regresión y 'c' si es clasificación, el valor del parámetro de regularización, el valor de la desviación estándar γ, y el tipo de kernel, que por defecto es un kernel Gaussiano.

2. Realice un modelo de regresión usando máquinas de soporte vectorial con un kernel Gaussiano. Cambie los valores del box constraint y gamma variando desde 0.01 hasta 100 en potencias de 10, y complete la siguiente tabla. La tabla 1 completela con el resultado de Y1 y la tabla 2 completela con los resultados de Y2. Haga las modificaciones necesarias para completarlas de forma correcta.

   Responda las siguientes preguntas:

   - Explique en sus palabras que son los vectores de soporte.

     R:/

   - Explique en sus palabras en que se diferencian las maquinas de soporte vectorial para clasificación con las maquinas de soporte para regresión.

     R:/

Cuadro 1: Problema de Regresion: Kernel Gaussiano Y1

Box Constraint	**Gamma**	**ECM**	**Intervalo de Confianza**
0.01	0.01		
	0.1		
	1		
	10		
	100		
0.1	0.01		
	0.1		
	1		
	10		
	100		
1	0.01		
	0.1		
	1		
	10		
	100		
10	0.01		
	0.1		
	1		
	10		
	100		
100	0.01		
	0.1		
	1		
	10		
	100		

Cuadro 2: Problema de Regresion: Kernel Gaussiano Y2

Box Constraint	**Gamma**	**ECM**	**Intervalo de Confianza**
0.01	0.01		
	0.1		
	1		
	10		
	100		
0.1	0.01		
	0.1		
	1		
	10		
	100		
1	0.01		
	0.1		
	1		
	10		
	100		
10	0.01		
	0.1		
	1		
	10		
	100		
100	0.01		
	0.1		
	1		
	10		
	100		

3. Implemente un modelo de SVM con un kernel lineal y la estrategia *One vs All* para realizar la clasificación. Cambie los valores del box constraint desde 0.01 hasta 100 en potencias de 10, y complete la tabla 3:

4. Repita el punto anterior para un kernel Gaussiano variando gamma desde 0.01 hasta 100 en potencias de 10, y complete la tabla 4:

   Responda las siguientes preguntas:

   - ¿Por qué es necesario utilizar la estrategia One vs All en este problema?

     R:/

   - Explique porque el valor del box constraint controla el número de vectores de soporte.

Cuadro 3: Problema de Clasificación: Kernel lineal

Box Constraint	**Eficiencia**	**Intervalo de Confianza**
0.01		
0.1		
1		
10		
100		

Cuadro 4: Problema de Clasificación: Kernel Gaussiano

Box Constraint	**Gamma**	**ECM**	**Intervalo de Confianza**
0.01	0.01		
	0.1		
	1		
	10		
	100		
0.1	0.01		
	0.1		
	1		
	10		
	100		
1	0.01		
	0.1		
	1		
	10		
	100		
10	0.01		
	0.1		
	1		
	10		
	100		
100	0.01		
	0.1		
	1		
	10		
	100		

Referencias

[1] Bishop, C.M. Pattern Recognition and Machine Learning. Springer, 2006.