Aprendizaje de Máquinas, Laboratorio #0

Discriminador Lineal

Fecha de entrega 30 de marzo de 2015, 23:59 hs

Nota: (1) Estas preguntas requieren pensar, pero no requieren largas respuestas. Por favor se tan conciso como sea posible. (2) Cuando envíes una pregunta al foro, por favor asegúrate de escribir el número de laboratorio y el número del problema, tal como L0 P2. (3) Para problemas que requieran programación, por favor incluye en tu envio el código (con comentarios) y cualquier figura que se haya solicitado graficar. Ten en cuenta que el código debe poder correr desde cualquier máquina (4) Si escribes tus soluciones a mano, por favor escribe claramente y utilizando una birome de color oscuro.

Se recomienda leer el apunte *Introducción al Aprendizaje de máquinas* para comprender las bases de la materia. En la sección 1.1 del apunte *Discriminadores Lineales* encontrará las bases teóricas necesarias y requeridas. A continuación comenzará la guía de trabajo de discriminantes lineales.

§ Problema 1. **[100 pts] Discriminador lineal para clasificación**

En este laboratorio se quiere aprender un clasificador que pueda determinar si una flor, de acuerdo a las medidas de pétalos y sépalos, pertenece o no a una determinada familia. Por ello trabajaremos con un conjunto de datos (dataset) `iris.data` que ha sido generado por un experto en el tema. Para ello utilizaremos un modelo de regresión lineal.

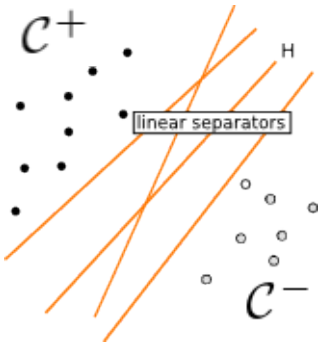

Figura 1: Gráfica de separadores lineales, posibles soluciones a un conjunto de datos

Los modelos de regresión lineal han sido principalmente utilizados en estadística durante los últimos 30 años. Estos modelos representan un problema de regresión a través de un conjunto de variables reales $X = \{X_1, X_2, \ldots, X_p\}$ utilizadas para predecir el valor de una variable real Y. De esta manera, dado un vector de *variables de entrada* $X^T = (X_1, X_2, \ldots, X_p)$, predecimos la *variable de salida* Y utilizando el siguiente modelo

$$\hat{Y} = \hat{w}_0 + \sum_{j=1}^{p} X_j \hat{w}_j, \tag{1}$$

donde $\hat{Y}$ denota la predicción para las variables Y y los $\hat{w}$ denotan un *conjunto de pesos* que deben aprenderse para ajustar el modelo a un problema específico. Por razones de conveniencia se agrega una

constante 1 en el vector X, incluyendo $\hat{w}_0$ en el vector de pesos $\hat{w}$, de esta manera podemos expresar (1) simplemente como un producto interior

$$\hat{Y} = X^T \hat{w}. \tag{2}$$

Para problemas de clasificación con dos clases, i.e. el dominio de la variable Y es 2, un enfoque es asignar a Y el dominio de valores $\{1, -1\}$ que denotan las dos posibles clases $C = \{C_1, C_2\}$ del problema. Entonces para las predicciones $\hat{Y}$ asignaremos la clase C_1 si $\hat{Y} > 0$ y C_2 si $\hat{Y} \leq 0$.

Para aprender el conjunto de pesos $\hat{w}$ utilizamos un conjunto de *datos de entrenamiento*, el cual está formado por N vectores de valores observados para las variables X, junto con las correspondientes clases para las variables Y, es decir, un conjunto de tuplas $\{(x_0, y_0), (x_1, y_1), \ldots, (x_N, y_N)\}$. Denotaremos como una matriz $\mathbf{X}$ de $N \times p$ a las observaciones de las variables X y como un vector columna $\mathbf{y}$ de $N \times 1$ a sus correspondientes clases. Para nuestro caso los pesos serán ajustados utilizando *mínimos cuadrados*, eligiendo aquellos que minimizan la *residual sum of squares*

$$RSS(w) = (\mathbf{y} - \mathbf{X}w)^T(\mathbf{y} - \mathbf{X}w). \tag{3}$$

El vector de pesos $\hat{w}$ que minimiza (3) se obtiene al derivar con respecto a w y despejar

$$\hat{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}, \tag{4}$$

(a) **[20 pts]** *Pre-procesamiento*

El pre-procesamiento consiste en preparar los datos para poder utilizarlos en las fórmulas numéricas.

El problema consiste en aprender un clasificador lineal (2) utilizando una matrix de entrada $\mathbf{X}$ (o dataset) almacenada en el archivo `iris.data`. Los atributos, representados en cada una de las columnas, son:

a) sépalo largo en cm
b) sépalo ancho en cm
c) pétalo largo en cm
d) pétalo ancho en cm
e) clase: cuyo valor puede ser *Iris Setosa*, *Iris Versicolour* o *Iris Virginica*

Dado que utilizaremos un clasificador de tipo binario y el dataset utiliza tres clases categóricas, debe pre-procesarlo para solamente considerar las siguientes clases: *Iris Virginica* y *No Iris Virginica*. La clase *Iris Virginica* debe sustituirse por el valor 1. Las restantes por el -1. Debe completar el código de la función pre_process que se encuentra en el template.

(b) **[20 pts]** *Aprendizaje* Utilizando este dataset pre-procesado, deberá completar la función learn para que lo lea y lo divida en dos conjuntos de datos: uno formado por el 70 % de los datos que serán el *conjunto de entrenamiento* y los restantes serán el *conjunto de testeo*. El primer conjunto deberá ser utilizado para aprender los pesos $\hat{w}$ siguiendo (4)

(c) **[30 pts]** *Evaluación* Luego utilizando el conjunto de testeo se deberá evaluar el error de clasificación E (función evaluate). Esta evaluación se realizará bajo el siguiente esquema: se tomarán un número $N \in \{50, 100, 150\}$ de datos del dataset y se ejecutarán $R = 10$ repeticiones donde en cada una

se obtendrán diferentes conjuntos de entrenamiento/testeo, para las cuales se calculará el error de clasificación.

Para las 10×3 ejecuciones se deberá reportar en un archivo denominado `outputs.raws` el número de datos N, número de repetición R y error E obtenido con el siguiente formato:

```
# Descripción de campos:
# nombre de dataset
# N
# número de repetición
# error de clasificación
iris.data,N,R,E
```

(d) **[30 pts]** *Post-procesamiento* Una vez obtenidos los datos para todas las ejecuciones deberá post-procesar el archivo `outputs.raws`, para ello deberá completar la función `pos_process` tome como parámetro un número N de datos y calcule el promedio μ para los errores E obtenidos en las 10 repeticiones. Debe eliminar de `outputs.raws` todos los renglones que comiencen con el caracter `#`; luego con los restantes renglones calcular los promedios μ; y finalmente reportar por pantalla la tupla (N, μ), donde cada elemento deberá estar separado por un tabulador y μ deberá imprimirse con dos cifras significativas.