Taller de Aprendizaje de Máquina: Estimación no Paramétrica

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

August 26, 2015

1 Marco teórico

Como se ha visto hasta ahora en clase, los dos problemas básicos de aprendizaje de máquina: el problema de clasificación y el problema de regresión, requieren ambos de la estimación de funciones de densidad de probabilidad. Las aproximaciones básicas de regresión polinomial y de estimación de valores medios y desviaciones estándar, hacen parte del conjunto de los modelos de aprendizaje paramétricos en los cuales se hacen suposiciones sobre la funciones de densidad de probabilidad (*fdp*) que representan el conjunto de datos de entrenamiento. En el otro costado se encuentran los modelos de aprendizaje no paramétricos, en los cuales no se hacen suposiciones sobre la forma de la *fdp* que representa los datos, aunque a diferencia de lo que puede pensarse por el nombre, dichos modelos requieren incluso de un número de parámetros mayor que el que se requiere en varios tipo de modelos paramétricos.

A continuación se van a presentar 3 métodos de estimación de *fdp* no parámetricos. Todos los métodos presentados permiten llevar a cabo tareas tanto de regresión como de clasificación. Es importante tener en cuenta que siempre partimos de un conjunto de entrenamiento $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ (estamos abordando tareas de aprendizaje supervisado), y dependiendo de la tarea necesitaremos llevar a cabo uno de los siguientes cálculos:

- **Regresión** Debemos encontrar un modelo $f(\mathbf{x})$ a partir de cual podamos calcular:

$$y = f(\mathbf{x}) = E[p(y|\mathbf{x})]$$

  En el caso discreto el valor esperado de la ecuación anterior se puede expresar como

$$E[p(y|\mathbf{x})] = \sum_{\forall y_i} y_i p(y_i|\mathbf{x}_i) = \sum_{\forall y_i} y_i \frac{p(\mathbf{x}_i, y_i)}{p(\mathbf{x}_i)}$$

- **Clasificación**

  A diferencia del caso anterior, en el problema de clasificación es necesario construir una *fpd* para cada clase a reconocer y el problema de encontrar la clase C a la cual pertenece un nuevo $\mathbf{x}^*$ se puede definir como:

  $$C = \max_y p(y|\mathbf{x}) = \max_y \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$$

  El problema de clasificación será abordado con detalle más adelante en el curso.

1.1 Método del histograma

Este es quizá el método de estimación no paramétrica más antiguo y más simple. En el método del histograma se divide el espacio de características en una malla y se contabiliza el número de datos que yacen en cada celda. Es necesario definir el ancho de las celda (por cada variable) y el punto donde comienza la división. Para efectos prácticos se puede definir el número celdas deseadas y el incremento se estima de acuerdo al rango dinámico de las variables en el conjunto de entrenamiento.

Algoritmo para el problema de regresión:

1. Definir el número de divisiones deseadas por cada variable N_c.
2. Encontrar el máximo y el mínimo de cada variable dentro del conjunto de entrenamiento.
3. Calcular el incremento para cada variable como $(\min - \max)/N_c$
4. Determinar los límites de los intervalos de acuerdo al incremento.
5. Crear una matriz H de igual número de dimensiones como variables (incluyendo la variable a predecir). Almacenar en H el número de y_i que contiene cada una de las celdas en las cuales se dividió el espacio.
6. Determinar un valor de y representativo para cada celda (valor medio entre los límites de la celda).
7. Para un nuevo $\mathbf{x}^*$ Estimar y^* como:

$$y^* = \sum_{l=1}^{N_c} y_l \frac{p(\mathbf{x}^8, y_l|H)}{p(\mathbf{x}^*|H)}$$

Para problemas de clasificación deberíamos definir para cada uno de los hipercubos en los que se divide el espacio, cuál es la clase más probable y para una muestra desconocida se le asignará la clase asociada al hipercubo en el cual se ubicó la muestra.

1.2 Método de los *k*-vecinos

En todos los métodos expuestos en este documento, la suposición fundamental es que el comportamiento esperado es similar dentro de una misma región, es decir, los cambios en la función de predicción son suaves. En este caso la estimación de la *fdp* no se realiza definiendo una región (hipercubo) representativa, sino, definiendo los puntos vecinos dentro del conjunto de entrenamiento a un nuevo punto $\mathbf{x}^*$ y el valor predicho será entonces el promedio de los valores y_i de los puntos vecinos a $\mathbf{x}^*$ [**?**].

Algoritmo para el problema de regresión:

1. Definir el número de vecinos a usar K.
2. Definimos una medida de distancia a utilizar, por ejemplo la distancia Euclidiana dada por:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^{m} (x_{il} - x_{jl})^2}$$

   donde x_{il} corresponde a la variable l del vector $\mathbf{x}_i$ que representa un elemento del conjunto de entrenamiento $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, y m es el número de variables.
3. Determinar los K vecinos más cercanos al punto $\mathbf{x}^*$ de acuerdo a la medida de distancia.
4. Calcular el correspondiente y^* como:

$$y^* = \frac{1}{K} \sum_{j=1}^{K} y_j^{\mathcal{V}}$$

   donde $y_j^{\mathcal{V}}$ corresponde al valor de y_j que acompaña al vector $\mathbf{x}_j$ considerado vecino de $\mathbf{x}^*$.

Para problemas de clasificación debemos determinar también los k vecinos más cercanos, y utilizamos la moda en lugar de la media para determinar el valor de la variable a predecir. Es decir, buscamos la clase que más se presenta dentro del conjunto de vecinos.

1.3 Método de ventana de Parzen o método Kernel

Finalmente el último de los tres métodos considerados en este documento, construye una función de predicción sin delimitar una región de vecindad o un número de vecinos, sino dandole un peso a cada uno de los puntos en el conjunto de entrenamiento [**?**]. Dicho peso será mayor cuanto más cercano esté el punto de entrenamiento al nuevo objeto $\mathbf{x}^*$ y se asigna a través de una función conocida como kernel. El cálculo de la probabilidad de una nueva muestra a partir de la función kernel se realiza así:

$$f(\mathbf{x}^*) = \frac{1}{N}\sum_{i=1}^{N} k\left(\frac{\|\mathbf{x}^* - \mathbf{x}_i\|}{h}\right)$$

Por consiguiente en un problema de clasificación se debe tener una función f por cada clase, estimar la probabilidad de una muestra $\mathbf{x}^*$ en cada clase y asignar la clase que mayor probabilidad entregue. Es decir que los $\mathbf{x}_i$ usados en cada f corresponden únicamente a muestras de una misma clase.

Por otro lado el algoritmo para el problema de regresión sería:

1. Definimos una medida de distancia a utilizar, por ejemplo la distancia Euclidiana.

2. Definir una función (kernel) que determine el peso de cada punto en el conjunto de entrenamiento, deacuerdo a su distancia con el nuevo objeto $\mathbf{x}^*$ para el cual se desea predecir un valor y^*. La función kernel más empleada es el kernel Gaussiano dado por:

   $$k(u) = \frac{1}{2}\exp\left(-\frac{1}{2}u^2\right)$$

   donde $u = d(\mathbf{x}_i, \mathbf{x}_j)/h$ y h es la ventana de suavizado.

3. Realice la predicción escogiendo alguna de las posibles funciones, por ejemplo:

   Nadaraya-Watson

   $$y = \frac{\sum_{i=1}^{N} k\left(d(\mathbf{x}^*, \mathbf{x}_i)/h\right) y_i}{\sum_{i=1}^{N} k\left(d(\mathbf{x}^*, \mathbf{x}_i)/h\right)} = \frac{\sum_{i=1}^{N} k\left(\|\mathbf{x}^* - \mathbf{x}_i\|/h\right) y_i}{\sum_{i=1}^{N} k\left(\|\mathbf{x}^* - \mathbf{x}_i\|/h\right)}$$

   Priestley-Chao

   $$y = \frac{1}{h}\sum_{i=2}^{N}(\|\mathbf{x}_i - \mathbf{x}_{i-1}\|)k(\|\mathbf{x}^* - \mathbf{x}_i\|/h)y_i$$

   donde $\|\cdot\|$ representa una norma o medida de distancia.

Nota. Recuerde que en los dos últimos métodos es necesario realizar una previa normalización de las variables, por cuanto ambos métodos requieren el cálculo de medidas de distancia que se ven afectadas debido al sesgo que puede introducir las diferencias en los rangos dinámicos de las variables utilizadas.

2 Ejercicios

1. Adjunto a este taller encontrará los archivos: *Main.m, normalizar.m, gaussianKernel.m, vecinosCercanos.m, ventanaParzenClass.m y ventanaParzenRegress.m*. El archivo *Main.m* es el script principal, desde el cual se ejecutan todas las instrucciones para realizar el taller. Una vez corra el Script principal se le solicitará ingresar el numeral del punto que desea resolver (es decir 3,4,5 o 6). Analice con cuidado el script y comprenda como esta construido.

2. * Genere un conjunto de 1000 muestras artificiales con una distribución que corresponda a la suma de dos Gaussiana con diferente media, ambas de una sola dimension. Grafique el histograma de los datos generados.

3. Describa la base de datos utilizada en el problema de regresión. Cuantas son las muestras de entrenamiento y validación y cuantas son las características.

   R/:

   Para poder resolver el problema de regresión con vecinos cercanos debe completar el archivo *vecinosCercanos.m*, el cual tiene indicado las lineas donde se debe implementar el cálculo para determinar los K vecinos cercanos de una muestra.

   Una vez haya completado el código, ejecute varias veces el proceso de entrenamiento y evaluación cambiando el parámetro k, el cual es el número de vecinos, y complete la siguiente tabla con los valores del error cuadrático medio (ECM) obtenidos:

Número de vecinos	Error Cuadrático Medio (ECM)
1	
2	
3	
4	
5	
6	
7	
100	

Responda las siguientes preguntas:

- ¿Cuál es el porcentaje de la base de datos usado en el conjunto de prueba?
  R/:

- ¿Por qué cree que se obtiene este resultado con 100 vecinos?
  R/:

4. Ahora resuelva el problema de regresión con el método de ventana de parzen. Para hacerlo debe completar el archivo *ventanaParzenRegress.m*, el cual tiene indicado las lineas donde se debe implementar la función de predicción de Nadaraya-Watson.

   Una vez haya completado el código, ejecute varias veces el proceso de entrenamiento y evaluación cambiando el parámetro h, donde h es el ancho de la ventana de suavizado, y complete la siguiente tabla con los valores del error cuadrático medio (ECM) obtenidos:

Ventana de suavizado	Error Cuadrático Medio (ECM)
0.05	
0.1	
1	
10	

Responda las siguientes preguntas:

- ¿Cuál es la formula usada para calcular el ECM?
  R/:

- ¿Por qué cree que se obtiene este resultado con h igual a 0.01?
  R/:

5. Describa la base de datos utilizada en los problemas de clasificación. Cuantas son las muestras de entrenamiento y validación, cuantas son las características y cuantas son las clases del problema y cuantas son las muestras de cada clase.

   R/:

   Para poder resolver el problema de clasificación con vecinos cercanos debe completar el archivo *vecinosCercanos.m*, el cual tiene indicado las lineas donde se debe implementar el cálculo para determinar los K vecinos cercanos de una muestra.

   Una vez haya completado el código, ejecute varias veces el proceso de entrenamiento y evaluación cambiando el parámetro k, el cual es el número de vecinos, y complete la siguiente tabla con los valores de eficiencia y error de clasificación obtenidos:

Número de vecinos	**Eficiencia**	**Error de clasificación**
1		
2		
3		
4		
5		
6		
7		

Responda las siguientes preguntas:

- ¿Por que se usa la moda en el caso de clasificación con el método de los K vecinos?
  R/:

- ¿Por qué cree que se deben armar los conjuntos de entrenamiento y prueba de forma aleatoria?
  R/:

6. Ahora resuelva el problema de clasificación con el método de ventana de parzen debe completar el archivo *ventanaParzenClass.m*, el cual tiene indicado las lineas donde se debe implementar el cálculo de la función de probabilidad.

   Una vez haya completado el código, ejecute varias veces el proceso de entrenamiento y evaluación cambiando el parámetro h y complete la siguiente tabla con los valores de eficiencia y error de clasificación obtenidos:

Ventana de suavizado	**Eficiencia**	**Error de clasificación**
0.05		
0.1		
1		
10		

Responda las siguientes preguntas:

- ¿Por qué el modelo de ventana de parzen es un modelo no paramétrico?
  R/: