BOOSTING

FUNDAMENTOS Y ALGORITMOS

1.1 Clasificacion de los Problemas y Machine Learning

Este libro se enfoca principalmente en el problema de clasificacion, en el cual, el objetivo es categorizar objetos dentro de relativamente un pequeño conjunto de clases. Por ejemplo, el sistema OCR debe clasificar imagenes de letras en las categorias A, B, C, etc. Otro ejemplo es el diagnostico medico para el problema de clasficacion en el que el objetivo es diagnosticar un paciente. El filtro spam es otro ejemplo de problema de clasificacion en el que se intenta categorizar un correo como spam o no.

Para solucionar el problema de clasificacion, nos enfocamos especialmente en el concepto de machine learning. El Machine Learning estudia el diseño de metodos automaticos para hacer predicciones del futuro basado en experiencias anteriores. En el contexto del problema de clasificacion, el metodo de machine learning intenta aprender para predecir la correcta clasificacion de ejemplos nuevos, a travez, de una examinacion cuidadosa de ejemplos que previamente han sido etiquetados con su correcta clasificacion, usualmente por un humano.

Nos referimos al objeto a clasificar como instancia. Por ejemplo en OCR, las instancias se describen como las imagenes de letras. Al espacio de todas las posibles instancias se llama instance space or domain y se denota por X.

Durante el entrenamiento, un algoritmo de aprendizaje recibe como entrada un training set etiquetados, denominado como trainig examples. La salida del algoritmo de aprendizeje es una regla de prediccion llamado como clasificador o hipotesis. Matematicamente un clasificador es una funcion que mapea una instancia a una etiqueta.

Para medir la cualidad de un clasificador, usamos el concepto de error rate, esto es, la frecuencia en que se realiza una correcta clasificacion, para ello, necesitamos un test set, un conjunto separado de ejemplos de prueba. El clasificador se evalua para cada instancia y sus predicciones se compara sobre el mismo test set. La fraccion de ejemplos que generan una incorrecta clasificacion se llama como test error del clasificador. Similarmente la fraccion de errores en el training set se denomina como training error y la fraccion de predicciones correctas se llama accuracy.

Por supuesto, el rendimiento de la clasificacion en el training set no es de mucho interes dado que el objetico es construir un clasificador que trabaja sobre datos nuevos. Por otro lado, si no hay relacion entre todos los elementos del training set y el test set entonces el problema de aprendizaje no tiene solucion, el futuro puede ser predicho solo si recalculamos el pasado.

La generalization error de un clasificador mide la probabilidad de una incorrecta clasificacion en una ejemplo aleatorio, desde la distribucion D, equivalentemente, el error de generalizacion es el error de test esperado del clasificador sobre cualquier test generado por D.

1.2 Boosting

Boosting asume la disposicion de una base o weak learning algorithm que dado una lista de ejemplos etiquetados, produce una base o weak classifier. El objetivo de boosting es mejorar el rendimiento de un weak learning algorithm mientras se trata como “black box” y puede llamarse repetidamente, como una subrutina. Esta suposición es, que la base aprende a producir una hipotesis debil que es un poco mejor que una busqueda aleatoria sobre cada ejemplo sobre el que se hizo el entrenamiento, se llama weak learning assumption.

Como las palabras classifier e hypothesis, usamos los terminos base y weak mas o menos intercambiables pero como con enfatizando el pesimo rendimiento y base connotandose como un building block.

Como cualquier algoritmo de aprendizaje, un algoritmo boosting toma como entrada un training set (x1, y1),...,(xm, ym) donde cada xi es una instancia de X y cada yi como una clase o etiqueta

La idea clave detras del boosting es escoger los training set para la base learner de tal modo como se fuerza a inferir algo nuevo de los datos cada vez que sea llamado. Esto puede ser logrado escogiendo sets training en el que nosotros rasonablemente esperamos un rendimiento pobre del clasificador.

Ahora estamos listos para describir en detalle al algoritmo boosting denominado AdaBoost, que incorpora estas ideas y cuyo pseudo codigo se muestra en la siguiente figura.

Ada Boost procede en rondas o llamadas iterativas a la base learner. Para escoger el training set que se enviara al base learner en cada ronda, AdaBoost mantiene una distribucion sobre los training examples, la distribucion usada en la t-th ronda se denota por Di y el peso se asigna a cada training example,

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathcal{X}$, $y_i \in \{-1, +1\}$.
Initialize: $D_1(i) = 1/m$ for $i = 1, \ldots, m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \mathcal{X} \to \{-1, +1\}$.
- Aim: select h_t to minimalize the weighted error:

$$\epsilon_t \doteq \mathbf{Pr}_{i \sim D_t}[h_t(x_i) \neq y_i].$$

- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$.
- Update, for $i = 1, \ldots, m$:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

$$= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t},$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \text{sign} \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right).$$

El trabajo del base learner es encontrar un clasificador ht (que va desde X hasta {-1,+1}) apropiado para la distribucion Dt. Acorde a lo mencionado anteriormente, la cualidad de una base classifier se mide por el error weighted dado por Dt.

$$\epsilon_t \doteq \mathbf{Pr}_{i \sim D_t}[h_t(x_i) \neq y_i] = \sum_{i : h_t(x_i) \neq y_i} D_t(i).$$

Aqui Pr(i-Dt) denota la probabilidad relativa con respecto a la selección aleatoria de un ejemplo acorde a una distribucion Dt. El error es medido con respecto a la misma distribucion Dt, cuyo clasificador base fue entrenado.

Si en un clasificador hace que cada una de sus predicciones sea completamente aleatoria, si se tiene que cada etiqueta pueda ser -1 o +1 con igual probabilidad, entonces la probabilidad de una clasificacion errornea en cualquier ejemplo es exactamente ½, entonces el error de este clasificador sera siempre ½ independientemente de los datos sobre el cual, el error se a medido. Entonces una weak hypothesis con error et = ½ se puede obtener trivialmente formulando cada prediccion como una suposicion aleatoria. La suposicion en cada weak learning para nuestros propositos actuales, establece una suposicion de que el error de cada clasificador essta bordeado lejos de ½, tal que et sea (1/2-y) para alguna constante positiva y.

Para los pesos Dt(i) que AdaBoost calcula sobre cada training example, en la practica, existen diversas vias en el que estos pueden ser usados por el base learner. En algunos casos, el base learner puede usar estos pesos directamente, en otros casos, un training set sin peso es generado por la base learner mediante la selección de ejemplos aleatorios a partir del original training set.

Regresando al ejemplo del filtro spam, las instancia xi corresponden a los mensajes de email y sus etiquetal yi, que representa la correcta clasificacion como spam o ham. El clasificador base son las reglas que se dan por el weak learning donde las subcolleciones en el que se ejecutan se escojen aleatoriamente acorde a la distribucion Dt.

Una vez obetenida l.a base ht, AdaBoost escoge un parametro alfa_t que indica la importancia asignado a ht.