Curso 2016-2017
Grado en Ingeniería Informática
Universidad de Granada

Cuestionario 3

Carlos Manuel Sequí Sánchez

29 de mayo de 2017

1. Considere un modelo de red neuronal con dos capas totalmente conectadas: d unidades de entrada, n_H unidades ocultas y c unidades de salida. Considere la función de error definida por $J(\mathbf{w}) \equiv \frac{1}{2}\sum_{k=1}^{c}(t_k - c_k)^2 = \frac{1}{2}||\mathbf{t} - \mathbf{z}||^2$, donde el vector $\mathbf{t}$ representa los valores de la etiqueta, $\mathbf{z}$ los valores calculados por la red y $\mathbf{w}$ los pesos de la red. Considere que las entradas a la segunda capa se calculan como $z_k = \sum_{j=0}^{N_H} y_j w_{kj} = \mathbf{w}_k^t \mathbf{y}$ donde el vector $\mathbf{y}$ representa la salida de la capa oculta.

   a) Deducir con todo detalle la regla de adaptación de los pesos entre la capa oculta y la salida.

   b) Deducir con todo detalle la regla de adaptación de los pesos entre la capa de entrada y la capa oculta.

   Usar θ para notar la función de activación.

2. Tanto "bagging" como validación-cruzada cuando se aplican sobre una muestra de datos nos permiten dar una estimación del error de un modelo ajustado a partir de dicha muestra de datos. Discuta cuál de los dos métodos considera que obtendrá una mejor estimación del error. Especificar con precisión las razones.

Fuente: Transparencias de teoría de la asignatura de Aprendizaje Automático (UGR).
La técnica del **Bagging** es una mezcla entre la técnica de Bootstrap y promedios:

- **Bootstrap:** consiste en realizar X conjuntos de datos de entrenamiento del mismo tamaño que el conjunto original, remuestreando de forma aleatoria y con reemplazamiento para obtener nuevas muestras, dando así la posibilidad de tener elementos repetidos dentro del conjunto Xi y, por tanto, pudiendo dejar muestras fuera de dicho subconjunto. Una vez obtenidos los dataset con los que vamos a entrenar nuestros modelos (conjunto de datos Train), obtenemos los conjuntos de datos de validación (los datos para el Test), los cuales serán todas aquellas muestras que no hayan sido escogidas en los subconjuntos del Train(por lo general 1/3 de los datos del conjunto inicial de datos son usados para el conjunto de datos Test). Uno de los problemas de esta técnica es que los árboles de decisión creados presentan una alta varianza y, es por ello por lo que usamos Bagging, para reducir esa gran varianza.

- **Promedios:** Con el fin expuesto, Bagging construye X árboles de regresión usando los X conjuntos de entrenamiento generados por Bootstrap y realiza el promedio de las predicciones resultantes. Al realizar dicho promedio (teniendo en cuenta que los árboles inicialmente no están podados) la varianza se reduce considerablemente, obteniendo una técnica con un cierto nivel en la relación sesgo-varianza.

La técnica de **validación cruzada** consiste en hacer X pares de conjuntos Train-Test de manera que utilicemos a la hora de validar una medida del error promedio con las sucesivas particiones creadas y obtener así una validación más fiable y ajustada que realizándola únicamente sobre un solo conjunto de datos. Los subconjuntos Train y Test creados en el caso de la validación cruzada son disjuntos, por lo que no tenemos datos repetidos entre ambos subconjuntos. Además cada par Train-Test de los X pares creados para la validación del modelo son distintos, de esta forma en cada par de conjuntos Test-Train creado tenemos que los subconjuntos Test son completamente distintos entre ellos, haciendo que la validación se realice cada vez con un subconjunto del dataset distinto, lo cual hace que el ajuste del error sea alto.
Para calcular el error simplemente se obtiene todos los $Eout_i$ correspondientes a cada par de particiones Test-Train y se calcula la media de dichos errores, lo que nos dará el E_{cv}

Como conclusión sobre la mejor de las estimaciones obtenidas comparando Cross Validation(CV) con Bagging, pienso que el hecho de que CV utilice todos y cada uno de los datos para aprender mediante distintos conjuntos Train en las sucesivas particiones, es un hecho relevante a la hora de estimar el mejor modelo para ajustarnos a la función buscada, cosa que Bagging no hace, ya que este último ofrece completa libertad para que en el conjunto de Train haya muestras repetidas, lo que produce que no se aprenda

con todos y cada una de las muestras del dataset como hace CV. Debido también a este motivo explicado, Bagging consta con un mayor poder de generalización, ya que ajusta menos el predictor a los datos del Train sin embargo, ni Bagging ni CV sufren los efectos del sobreajuste, por lo que no es un motivo para decantarse por una u otra técnica.
Bagging aumenta el ratio de predicción al tener varios árboles en contraste con un solo árbol.
En definitiva, y como última diferencia, pienso que no conviene utilizar Bagging frente a CV por el hecho de que, aun siendo un poco mejor a la hora de la predicción, la interpretación de los resultados es muy complicada, por tanto, teniendo en cuenta todo lo comentado hasta el momento pienso que la mejor estimación del error la hará la técnica de Validación Cruzada.

3. Considere que dispone de un conjunto de datos linealmente separable. Recuerde que una vez establecido un orden sobre los datos, el algoritmo perceptron encuentra un hiperplano separador interando sobre los datos y adaptando los pesos de acuerdo al algoritmo

Algorithm 1 Perceptron

Entradas: $(\mathbf{x}_i, y_i), i = 1, \ldots, n$, $\mathbf{w} = 0$, $k = 0$
repeat
    $k \leftarrow (k + 1) \bmod n$
    if $\text{sign}(y_i) \neq \text{sign}(\mathbf{w}^T \mathbf{x}_i)$ **then**
        $\mathbf{w} \leftarrow \mathbf{w} + y_i \mathbf{x}_i$
    end if
until todos los puntos bien clasificados

Modificar este pseudo-código para adaptarlo a un algoritmo simple de SVM, considerando que en cada iteración adaptamos los pesos de acuerdo al caso peor clasificado de toda la muestra. Justificar adecuadamente/matematicamente el resultado, mostrando que al final del entrenamiento solo estaremos adaptando los vectores soporte.

```
Repeat
{
  //Buscamos el punto peor clasificado
  for (i in cantidadPuntos)
  {
    K ← (K+1) mod n
    if ( yi * (w^T xi + b) ≤ 0) //Punto mal clasificado
    {
      dist = calcularDistanciaDesdePuntoHastaHiperplano();
      if (dist > maximaDistanciaActual)
      {
        maximaDistanciaActual = dist;
        //almacenamos ese punto con su etiqueta
        peorX = xi;
        peorY = yi;
      }
    }
  }
  //Actualizamos los pesos teniendo en cuenta al peor
  // clasificado.
  w = w + peorY * peorX;
} Hasta que todos los puntos estén bien clasificados;
```

- El cálculo de la distancia desde un punto hasta un hiperplano se realiza siguiendo esta fórmula:

$$d(x_i, h) = \frac{y_i (w^T x_i + b)}{\|w\|}$$

4. Considerar un modelo SVM y los siguientes datos de entrenamiento: Clase-1:{(1,1),(2,2),(2,0)}, Clase-2:{(0,0),(1,0),(0,1)}
   a) Dibujar los puntos y construir por inspección el vector de pesos para el hiperplano óptimo y el margen óptimo.
   b) ¿Cuáles son los vectores soporte?
   c) Construir la solución en el espacio dual. Comparar la solución con la del apartado (a)

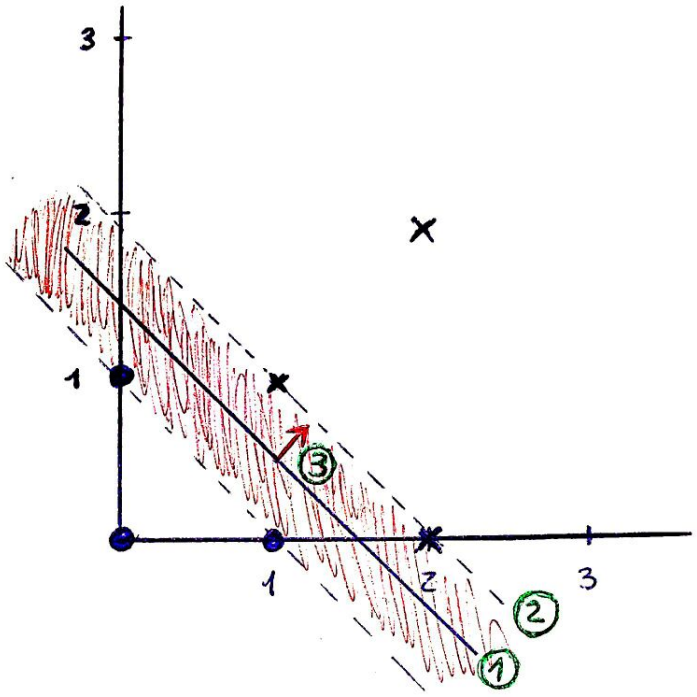


Figura 0.1: **Apartado A**

APARTADO B:
Vectores de soporte:

- **Clase 1:** {(1,0),(0,1)}
- **Clase 2:** {(2,0),(1,1)}

APARTADO C:

5. Una empresas esta valorando cambiar su sistema de proceso de datos, para ello dispone de dos opciones, la primera es adquirir dos nuevos sistemas idénticos al actual a 200.000 euros cada uno, y la segunda consiste en adquirir un sistema integrado por 800.000 euros.

   Las ventas que la empresa estima que tendrá a lo largo de la vida útil de cualquiera de sus equipos son de 5.000.000 de euros en el caso de positivo, a lo que la empresa le asigna una probabilidad de que suceda del 30 %, en caso contrario, las ventas esperadas son de 3.500.000 euros. ¿Que opción debería de tomar la empresa?

Debido a la ambigüedad del enunciado, procedo a dar mis dos propias interpretaciones sobre el mismo.
Interpretación del enunciado 1 Considero que los dos sistemas que cuestan 200.000 euros cada uno son independientes, por lo que cada uno de ellos puede llegar a generar 5.000.000 de euros en caso de funcionar con el máximo rendimiento. El sistema integrado, tal y como dice el enunciado, también puede llegar a generar 5.000.000 de euros en el caso positivo.
Una vez realizada la interpretación es trivial llegar a la conclusión óptima para las ganancias de la empresa:

- Por un lado tenemos que las dos máquinas idénticas a la actual pueden llegar a generar a la empresa un beneficio máximo de 10.000.000 de euros en caso de que ambas tengan una máxima productividad. En el caso opuesto, nos encontramos con que ambas máquinas trabajan a mínimo rendimiento y generan cada una 3.500.000 euros (7.000.000 euros en total).

- En el caso del sistema integrado tenemos que a máximo rendimiento, tal como expresa el enunciado puede generar 5.000.000 euros, mientras que a mínimo rendimiento puede generar 3.500.000 euros.

A la vista de la información expuesta, está claro que es preferible comprar dos máquinas por 400.000 euros para obtener una ganancia mayor (en cualquiera de los casos) que comprando una sola máquina por 800.000 euros con la cual es posible que la empresa no gane ni la mitad que con los dos sistemas.
Reiterando un poco lo afirmado, cabe destacar que en caso de comprar las dos máquinas y que trabajen a mínimo rendimiento la empresa obtendría más ganancias (7 millones de euros) que si comprase sel sistema integrado y éste trabajase a máximo rendimiento (5 millones de euros).

Interpretación del enunciado 2:
En la siguiente página muestro la resolución del ejercicio con la segunda interpretación, en la cual considero que los dos sistemas pueden obtener de forma conjunta una ganandia de 5.000.000 euros (en lugar de 10.000.000) y que si una sola falla, entonces la ganancia es de 3.500.000 euros.

costo: 400.000 €

Máquina 1 | Máquina 2

costo: 800.000 €

Máquina 3

- Probabilidad de que un equipo falle = 0'7
- Probabilidad de que un equipo funcione = 0'3

- Ganancias para el caso positivo = 5.000.000 €
- Ganancias para el caso negativo = 3.500.000 €

Caso 1

- Probabilidad de que M1 y M2 funcionen $(P\oplus(M_1 \wedge M_2)) = (0'3)^2 = 0'09$
- Probabilidad de que M3 funcione $(P\oplus(M3)) = 0'3$

si tenemos que ganancia (x) = $\frac{\text{prestaciones}}{\text{coste}}$

$$g(M1 \wedge M2) = \frac{0'09 \times 5000.000}{400\,000} = \boxed{1'125}$$

$$g(M3) = \frac{0'3 \times 5000\,000}{800.000} = \boxed{1'875}$$

En caso de que todas funcionen nos conviene comprar la máquina 3.

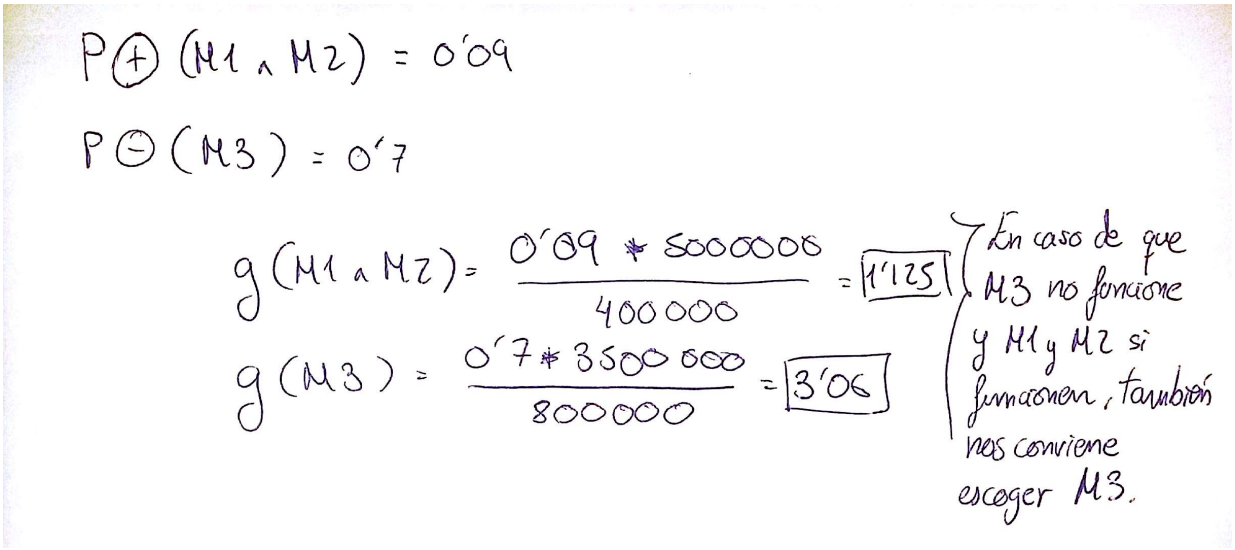
$$[P\oplus(M_1) \wedge P\ominus(M2)] \vee [P\ominus(M_1) \wedge P\oplus(M_2)] = 0'3 * 0'7 = 0'21$$

$P\ominus(M3) = 0'7$

$$g(M1 \wedge M2) = \frac{0'21 \cdot 3500.000}{400000} = \boxed{1'8}$$

$$g(M3) = \frac{0'7 \cdot 3500000}{800000} = \boxed{3'06}$$

En caso de que solo una de las 2 máquinas simples NO funcione y de que la M3 tampoco funcione, nos sigue conviniendo escoger la M3.



6. El método de Boosting representa una forma alternativa en la búsqueda del mejor clasificador respecto del enfoque tradicional implementado por los algoritmos PLA, SVM, NN, etc. a) Identifique de forma clara y concisa las novedades del enfoque; b) Diga las razones profundas por las que la técnica funciona produciendo buenos ajustes (no ponga el algoritmo); c) Identifique sus principales debilidades; d) ¿Cuál es su capacidad de generalización comparado con SVM?

APARTADO A:

La principal novedad ofrecida por dicho enfoque es que pretende conseguir un clasificador robusto (un buen clasificador) a partir de clasificadores debiles (malos, pero levemente mejores que los clasificadores aleatorios). Tambien es una novedad el hecho de que Boosting trabaja sobre el propio conjunto entero de entrada, no realiza particiones sobre el dataset para efectuar la clasificación. El algoritmo no se basa unicamente en la etiqueta de los datos para clasificarlos, sino en la distribución de probabilidad que tiene cada dato asociada para determinar la posibilidad de ser de una clase o de otra.

APARTADO B:

El concepto es sencillo: generamos malos clasificadores y juntamos sus resultados para obtener un buen clasificador. A partir de esta idea creo que es fácil pensar en que, lo que no tiene un clasificador, lo puede tener el otro y, a la hora de juntarlos, es posible conseguir un buen resultado debido a esa complementación entre modelos. Además, con el fin de generar distintos modelos de manera que cumplan esa complementacion mencionada entre modelos, los pesos asociados a los datos son manipulados en las sucesivas iteraciones, dando mayor prioridad a los elementos mal clasificados para que en las siguientes predicciones tengan mayor probabilidad de ser bien clasificados.

APARTADO C:

El mayor inconveniente para el método Boosting es el mal comportamiento ofrecido a la hora de soportar el ruido aleatorio en la clasificación, el cual esta presente en la mayoría (si no en todos) de los problemas de la vida real y por ello hace que su uso sea puesto en duda a la hora de la verdad.
Fuente:[1]
APARTADO D:

7. ¿Cuál es a su criterio lo que permite a clasificadores como Random Forest basados en un conjunto de clasificadores simples aprender de forma más eficiente? ¿Cuales son las mejoras que introduce frente a los clasificadores simples? ¿Es Random Forest óptimo en algún sentido? Justifique con precisión las contestaciones.

APARTADO A:
Pienso que el factor que mayormente contribuye a la eficiencia de Random Forests es el hecho de que utiliza una cantidad m de atributos (predictores) igual a $\sqrt{p}$, siendo p el número de atributos en la técnica de Bagging. Debido a esto, es decir, construyendo árboles que tan sólo usan m variables seleccionadas de forma aleatoria del conjunto de p predictores, introduce una cierta aleatoriedad que lleva a la disminución de la varianza.
Desde el punto de vista de la carga soportada por el sistema utilizado a la hora de aprender, Random Forests genera mas árboles pero de menor tamaño (menor numero de variables), lo que hace que el hecho de aprender sea mucho mas sencillo que aprender con un solo árbol de muchas más variables.

APARTADO B:
En bagging está presente la existencia de variables correladas debido a que todos los árboles construidos por esta técnica serán parecidos, en consecuencia la varianza no desciende lo deseado. Random Forests aporta la solución a este problema, haciendo que los árboles que construye Bagging no estén correlados, de esta manera sí habrá una mayor reducción de la varianza respecto del método de Bagging.
También introduce aún una mayor aleatoriedad que Bagging, debido a que escoge el subconjunto de m variables del conjunto de atributos p y, además, es capaz de dar estimaciones acerca de las variables con mayor importancia en la clasificación.

APARTADO C:
Pienso que no es un método de aprendizaje óptimo, puesto que está comprobado que con algunos grupos de datos Random Forests realiza sobreajuste sobre tareas de clasificación/regresión ruidosas. Fuente:[2]
Otro de los posibles fallos es que, el hecho de que tan solo se escoge un subconjunto **m** de los **p** atributos disponibles para aprender los modelos, me induce a pensar en que quizás el aprendizaje no es tan completo como el realizado por Bagging, el cual sí utiliza los p predictores disponibles.

Referencias

[1] Principal debilidad de boosting (artículo de la referencia nº 15 de los artículos referenciados al final de la página). `https://es.wikipedia.org/wiki/Boosting`.

[2] Principal debilidad de random forests (artículo de la referencia nº 8 de los artículos referenciados al final de la página). `https://es.wikipedia.org/wiki/Random_forest`.