VI Congreso Galego de Estatística e Investigación de Operacións
Vigo 5–7 de Novembro de 2003

BOOSTING CON REDES NEURONALES RBF. ANÁLISIS SESGO-VARIANZA EN UN PROBLEMA DE CLASIFICACIÓN

José María Matías Fernández[1]

[1]Departamento de Estadística e Investigación Operativa
Universidad de Vigo

RESUMEN

En el presente trabajo, se evalúa el comportamiento del *boosting* (AdaBoost) con redes RBF en el problema de clasificación. La conclusión principal del estudio es que las mejoras de la capacidad predictiva producidas por AdaBoost con respecto a la hipótesis base, se deben fundamentalmente a la reducción del sesgo. Además, el boosting resulta competitivo cuando la hipótesis base utilizada posee un nivel medio de complejidad.

Palabras e frases chave: Boosting, clasificación, redes neuronales, sesgo-varianza.
Clasificación AMS: 62Gxx, 62H30, 68T10, 68W40.

1. INTRODUCCIÓN

El análisis sesgo-varianza es una herramienta fundamental para comprender el comportamiento de cualquier algoritmo de estimación. Dicho análisis resulta una técnica natural en el problema de regresión con pérdida cuadrática debido a las propiedades de esta pérdida. En relación con el problema de clasificación con pérdida 0-1, se han propuesto diferentes descomposiciones del error de predicción tratando de emular la descomposición sesgo-varianza del problema de regresión minimocuadrática. Sin embargo, sólo recientemente ha surgido una propuesta unificadora inspirada en la filosofía utilizada por la descomposición tradicional.

En el presente trabajo se utiliza esta descomposición sesgo-varianza en el problema de clasificación para analizar el comportamiento del booosting (AdaBoost, [Schapire et al., 1998]). Para ello, se utlizan como hipótesis base redes neuronales RBF de diversos grados de complejidad.

2. SESGO Y VARIANZA EN UN PROBLEMA DE CLASIFICACIÓN

Sea el problema de clasificación con dos clases con codificación $\mathcal{Y} = \{-1, 1\}$ y con pérdida 0-1, $\ell(y, g(\mathbf{x})) = 1_{\{y \neq g(\mathbf{x})\}}$ y riesgo $R(g(\mathbf{x})) = \mathbb{E}_{\mathsf{Y}/\mathbf{x}}[\ell(\mathsf{Y}/\mathbf{x}, g(\mathbf{x}))]$, en el cuál, la predicción óptima viene dada por la regla de Bayes:

$$\mathsf{g}(\mathbf{x}) = \arg \inf_{g(\mathbf{x})} \{R(g(\mathbf{x})) = E_{\mathsf{Y}/\mathbf{x}}[\ell(\mathsf{Y}/\mathbf{x}, g(\mathbf{x}))]\}$$

TEOREMA 1. [Domingos, 2000] *En el problema de clasificación anterior, si $\mathcal{D}$ es un conjunto de muestras aleatorias de entrenamiento obtenidas de la distribución $\mathsf{P}_{\mathsf{X},\mathsf{Y}}$, y $\hat{g}_n$ es*

el estimador obtenido a partir de una muestra aleatoria $\mathsf{Z}^n = (\mathsf{Z}_1, ..., \mathsf{Z}_n)$, con $\mathsf{Z}_i = (\mathsf{X}_i, \mathsf{Y}_i)$, $i = 1 : n$, *se verifica la siguiente descomposición de la esperanza del error de predicción*:

$$\mathrm{MPE}(\hat{g}_n(\mathbf{x})) = E_{\mathsf{Z}^n,\mathsf{Y}/\mathbf{x}}[\ell(\mathsf{Y}/\mathbf{x}, \hat{g}_n(\mathbf{x}))] = c_1\sigma^2(\mathsf{Y}/\mathbf{x}) + \mathrm{Sesgo}(\hat{g}_n(\mathbf{x})) + c_2\,\mathrm{Var}(\hat{g}_n(\mathbf{x})) \quad (1)$$

donde:

$$c_1 = c_1(\mathbf{x}) = 2\mathbb{P}_{\mathsf{Z}^n}[\hat{g}_n(\mathbf{x}) = \mathsf{g}(\mathbf{x})] - 1$$
$$c_2 = c_2(\mathbf{x}) = \begin{cases} +1 & \text{si } \hat{g}^*_{\mathcal{D}}(\mathbf{x}) = \mathsf{g}(\mathbf{x}) \\ -1 & \text{en otro caso} \end{cases}$$

son constantes dependientes de $\mathbf{x}$ *(y también del algoritmo), y:*

$$\mathrm{Sesgo}(\hat{g}_n(\mathbf{x})) \doteq \ell(\mathsf{g}(\mathbf{x}), \hat{g}^*_{\mathcal{D}}(\mathbf{x}))$$
$$\mathrm{Var}(\hat{g}_n(\mathbf{x})) \doteq \mathbb{E}_{\mathsf{Z}^n}[\ell(\hat{g}^*_{\mathcal{D}}(\mathbf{x}), \hat{g}_n(\mathbf{x})]$$
$$\sigma^2(\mathsf{Y}/\mathbf{x}) \doteq \mathbb{E}_{\mathsf{Y}/\mathbf{x}}[\ell(\mathsf{Y}/\mathbf{x}, \mathsf{g}(\mathbf{x}))]$$

donde $\hat{g}^*_{\mathcal{D}}$ es la **predicción principal** *en* $\mathcal{D}$ *definida como*:

$$\hat{g}^*_{\mathcal{D}}(\mathbf{x}) \doteq \arg\min_{g(\mathbf{x})} E_{\mathsf{Z}^n}[\ell(g(\mathbf{x}), \hat{g}_n(\mathbf{x}))]$$

que, en el problema de clasificación, es la moda de las predicciones de las hipótesis entrenadas con muestras de $\mathcal{D}$.

Análogamente, puede definirse:

$$\mathrm{ME}(\hat{g}_n(\mathbf{x})) = \mathrm{Sesgo}(\hat{g}_n(\mathbf{x})) + c_2\,\mathrm{Var}(\hat{g}_n(\mathbf{x})) \quad (3)$$

y su correspondiente valor medio (análogamente para MPE):

$$\mathrm{ME}(\hat{g}_n) = \mathbb{E}_{\mathsf{X}}[\mathrm{ME}(\hat{g}_n(\mathsf{X}))] = \mathbb{E}_{\mathsf{X}}[\mathrm{Sesgo}(\hat{g}_n(\mathsf{X}))] + \mathbb{E}_{\mathsf{X}}[c_2(\mathsf{X})\,\mathrm{Var}(\hat{g}_n(\mathsf{X}))] \quad (4)$$

A diferencia de la descomposición en regresión, el término de varianza, puede resultar negativo debido a que $c_2(\mathbf{x}) = -1$ en los puntos donde existe sesgo: $\mathrm{Sesgo}(\hat{g}_n(\mathbf{x})) = \ell(\mathsf{g}(\mathbf{x}), \hat{g}^*_{\mathcal{D}}(\mathbf{x})) = 1$. Ello significa que **en estos puntos, cuanto más varianza menor error de generalización**. Claramente, esto no ocurre en el caso de regresión con pérdida cuadrática.

Por otra parte, es peculiar el término del ruido, pero su expresión sigue rigurosamente el paralelismo con la descomposición con pérdida cuadrática, [Domingos, 2000]. Para la regla de Bayes, su valor resulta intuitivo al ser el riesgo de Bayes en $\mathbf{x}$:

$$c_1(\mathbf{x})\mathbb{E}_{\mathsf{Y}/\mathbf{x}}[\ell(\mathsf{Y}/\mathbf{x}, \mathsf{g}(\mathbf{x}))] = (2\mathbb{P}_{\mathsf{Z}^n}[\mathsf{g}(\mathbf{x}) = \mathsf{g}(\mathbf{x})] - 1)\mathbb{E}_{\mathsf{Y}/\mathbf{x}}[\ell(\mathsf{Y}/\mathbf{x}, \mathsf{g}(\mathbf{x}))] = \mathbb{E}_{\mathsf{Y}/\mathbf{x}}[\ell(\mathsf{Y}/\mathbf{x}, \mathsf{g}(\mathbf{x}))]$$

Sin embargo, para cualquier otro algoritmo entrenado con muestras de $\mathcal{D}$, dicho valor resulta menor que dicho riesgo si $\mathbb{P}_{\mathsf{Z}^n}[\hat{g}_n(\mathbf{x}) = \mathsf{g}(\mathbf{x})] < 1$ e, incluso negativo, si $\mathbb{P}_{\mathsf{Z}^n}[\hat{g}_n(\mathbf{x}) = \mathsf{g}(\mathbf{x})] < 0.5$ con lo que, además, depende del algoritmo utilizado. Así, **si el estimador está lejos de ser óptimo**, **cuanto mayor ruido, mejor es su comportamiento**.

[Domingos, 2000] estudia el comportamiento de su propuesta de descomposición con árboles de regresión (*regression trees*) y con clasificadores *k-nearest neighbors*, y anima a estudiar dicho comportamiento con otros estimadores.

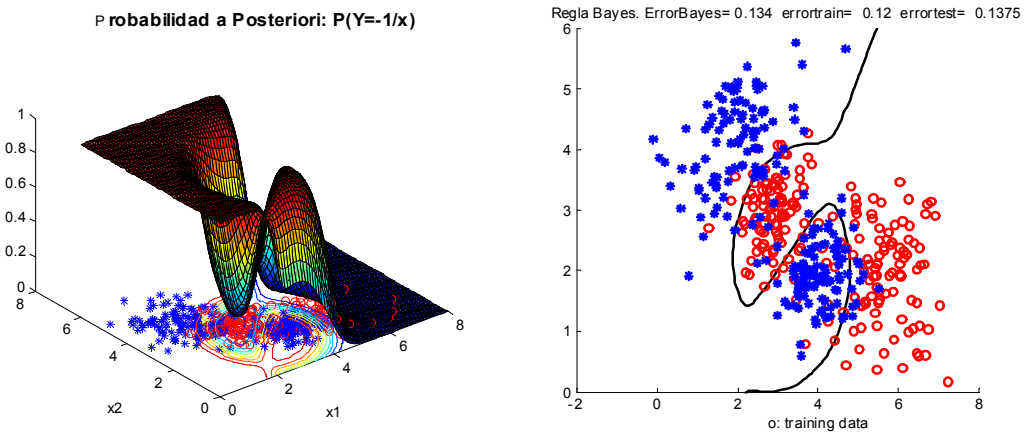

Figure 1: Izquierda: $\mathbb{P}(\mathsf{Y} = -1/\mathbf{x})$. Derecha: frontera de Bayes $\mathbb{P}(\mathsf{Y} = -1/\mathbf{x}) = 1/2$.

El problema de clasificación que hemos simulado, es un problema de dos clases, cada una de ellas constituida por dos *clusters* gausianos con la siguiente distribución de mixtura en cada clase (Figura 1):

$$\begin{array}{c} \mathsf{p}(\mathbf{x}/\mathsf{Y}=-1) = \frac{1}{2}\mathsf{p}(\mathbf{x};\mu_{11},\Sigma_1) + \frac{1}{2}\mathsf{p}(\mathbf{x};\mu_{12},\Sigma_2) \\ \mathsf{p}(\mathbf{x}/\mathsf{Y}=1) = \frac{1}{2}\mathsf{p}(\mathbf{x};\mu_{21},\Sigma_2) + \frac{1}{2}\mathsf{p}(\mathbf{x};\mu_{22},\Sigma_1) \\ \mu_{11} = (2,4)^t,\ \mu_{12} = (4,2)^t; \qquad \Sigma_1 = \mathrm{diag}(0.75, 0.75) \\ \mu_{21} = (3,3)^t,\ \mu_{22} = (5.5,2)^t; \quad \Sigma_2 = \mathrm{diag}(0.5, 0.5) \end{array} \tag{5}$$

donde $\mathsf{p}(\mathbf{x};\mu,\Sigma)$ es la densidad $N(\mu,\mathbf{\Sigma})$. Para la simulación, hemos considerado un conjunto $\mathcal{D}$ de 100 muestras de entrenamiento, de tamaño $n = 100$ cada una, $\mathcal{D} = \{(\mathbf{z}^n)_i,\ i = 1,...,100\}$ con igual número de datos en cada clase generados aleatoriamente según (5). Para este conjunto de muestras, hemos estimado la expresión (4), mediante sus correspondientes valores empíricos en una muestra de test de tamaño $n_t = 400$.

3. ADABOOST CON REDES RBF-OLS DE RADIO CONSTANTE

En nuestro conocimiento, los únicos experimentos de aplicación del *boosting* a las redes RBF son [Rätsch et al., 1999] y [Rätsch et al., 2001], en los que se entrenan dichas redes mediante el algoritmo del gradiente conjugado (algoritmo mixto). En estos experimentos, las redes RBF individuales presentaron un comportamiento sólo ligeramente inferior al *boosting*.

Las redes RBF así entrenadas tienden a poseer una alta componente de varianza, debido a la aleatoriedad del algoritmo de optimizacion no lineal empleado, por lo que es interesante conocer si otros algoritmos RBF más estables podrían competir mejor con el *boosting* o si éste puede aportar mejoras adicionales a los mismos.

Además, entre sus conclusiones generales, [Rätsch et al., 2001] destacan que, el *boosting* tiende a producir un salto en el valor mínimo de la distribución muestral del margen $\rho(\mathbf{x}_i) = y_i\hat{g}_n(\mathbf{x}_i)$, $i = 1:n$, y que, dicho valor mínimo, tiende a ser menor con hipótesis base menos *complejas*.

La figura 2 muestra los resultados 5000 iteraciones de AdaBoost sobre una red RBF-OLS con radio constante $\sigma = 10$. La figura recoge, de izquierda a derecha y de arriba a abajo:

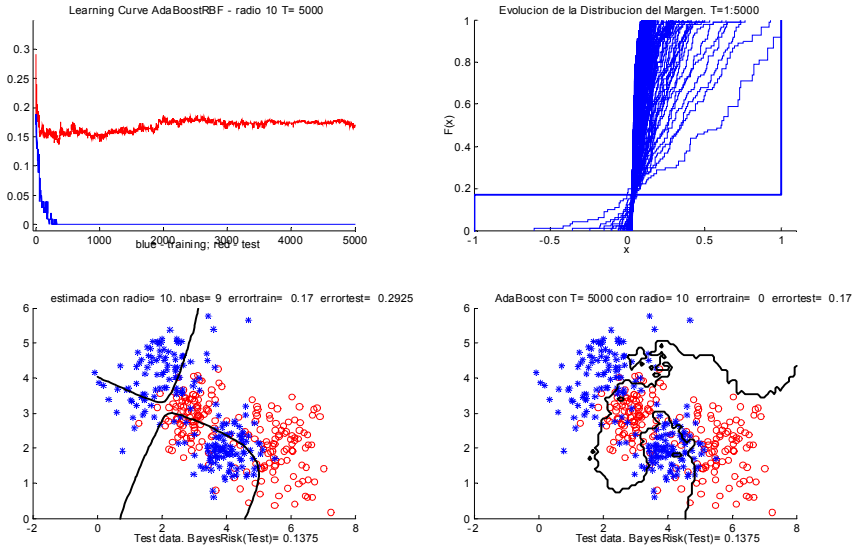

Figure 2: De izqda. a dcha. y de arriba a abajo: 1) curva de aprendizaje en entrenamiento (linea inferior) y en test (linea superior), 2) evolución de la distribución del margen (la línea gruesa es la distribución inicial), 3) clasificador inicial, 4) clasificador final.

1. La curva de aprendizaje de AdaBoost en la muestra de entrenamiento (línea inferior) y en la muestra de test (línea superior).

2. La evolución de la distribución del margen en la muestra de entrenamiento. La línea horizontal corresponde a la distribución del margen para $t = 1$ (la RBF inicial) donde se produce un 17% de errores en la muestra de entrenamiento. En una segunda fase, el algoritmo tiende a un margen mínimo positivo (que se consigue cuando la muestra de entrenamiento se clasifica sin errores, alrededor de la iteración $t = 300$) pero a costa de reducir el margen de los ejemplos mejor clasificados. En una tercera fase, dicho mínimo se mantiene pero siguen empeorando los mayores márgenes, situación que ya indica sobreajuste.

   Este comportamiento (y el pequeño margen mínimo obtenido con este clasificador base poco complejo –σ grande) es coherente con los resultados de [Rätsch et al., 2001], que denominan, a los puntos de margen mínimo, los vectores soporte, en analogía con el papel que juegan sus homónimos en las *support vector machines*.

3. La frontera de decisión producida por una RBF-OLS individual (izqda.), (error en test 0.2928), y la frontera de decisión producida después de la iteración 5000 de AdaBoost (error en test 0.1700).

Los gráficos anteriores muestran que, el éxito de AdaBoost, reside en la consecución de un gran margen mínimo para los datos de entrenamiento. Sin embargo, si los datos están sujetos a ruido, dicho objetivo puede llegar a ser contraproducente y producir sobreajuste.

En este contexto, interesa conocer el grado de complejidad adecuado para que el *boosting* produzca sus mejores resultados, suponiendo que algoritmo es detenido a tiempo antes del

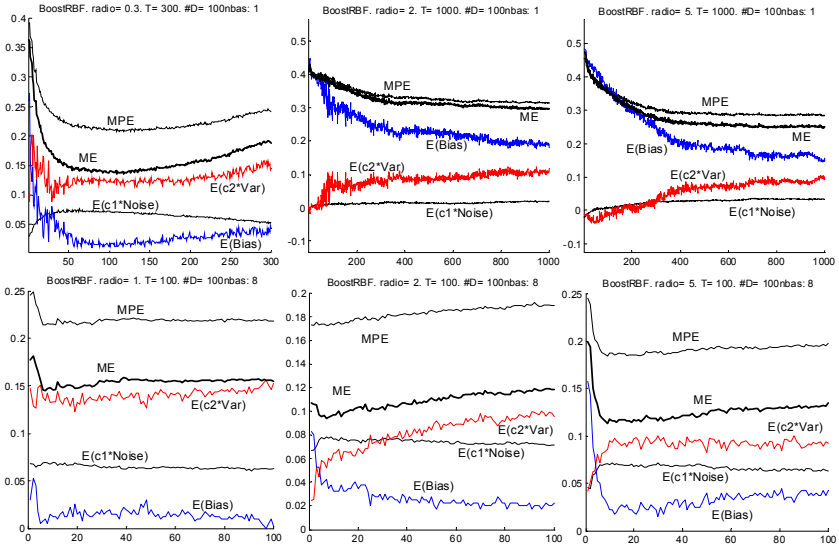

Figure 3: Evolución de la descomposición sesgo-varianza de AdaBoost aplicado a una función basica radial gausiana (arriba) y a 8 funciones básicas (abajo), para $\sigma = 0.3$, 2 y 5.

sobreajuste (por ejemplo con una muestra de validación).

Con este fin, hemos aplicado AdaBoost a diferentes redes RBF-OLS de radio constante, seleccionando los centros de entre los puntos de la muestra. La Figura 4 muestra la descomposición sesgo-varianza obtenida para diferentes valores del radio (σ) y diferente número de funciones básicas. Dicha tabla incluye, como comparación, los resultados de cada hipótesis base individual (H. Base), así como de dos redes RBF-OLS con radios con radios $\sigma = 2$ y 5, que utilizan el máximo número de funciones básicas permitido por la precisión del ordenador.

La figura 3 muestra la evolución de la descomposición sesgo-varianza de AdaBoost aplicado a redes RBF-OLS con una y ocho funciones básicas para valores $\sigma = 0.3$, 2.1 y 5.

4. CONCLUSIONES

En términos generales, sobre las pruebas realizadas, puede concluirse lo siguiente:

1. AdaBoost mejora siempre, sensiblemente, los resultados de las hipótesis individuales, sin embargo, la mejora con respecto a la mejor red RBF-OLS, es pequeña.

2. Dicha mejora es debida, en todos los casos, a una disminución radical del sesgo. Dicha disminución va acompañada de un incremento de varianza si el nivel de varianza del estimador base lo admite. Pero, si éste posee ya alta varianza, AdaBoost la reduce también aunque más ligeramente que el sesgo (Figura 4: casos con $\sigma = 0.3$ y número de básicas igual a 3, 5 y 8).

   En todo caso, si la varianza es pequeña, AdaBoost suele encontrar espacio para alguna mejora con respecto a la hipótesis base, siempre que, a su vez, el sesgo no sea excesivamente pequeño. Así, **la mejor combinación de circunstancias parece ser una varianza pequeña y un sesgo intermedio-bajo**.

AdaBoost-RBF	■	NB	Topt	E(Sesgo)	E(c2Var)	ME	E(c1Ru.)	MPE
H. Base	0.3	1		0.2950	0.0609	0.3559	0.0335	0.3894
			118	0.0150	0.1212	0.1362	0.0707	0.2069
H. Base		3		0.1200	0.1509	0.2709	0.0460	0.3169
			57	0.0150	0.1218	0.1368	0.0653	0.2020
H. Base		5		0.0725	0.1471	0.2196	0.0575	0.2771
			24	0.0225	0.1168	0.1393	0.0635	0.2027
H. Base		8		0.0325	0.1435	0.1760	0.0681	0.2441
			7	0.0100	0.1348	0.1448	0.0697	0.2145
H. Base	2	1		0.3975	-0.0056	0.3919	0.0043	0.3962
			+876	0.1825	0.1117	0.2942	0.0176	0.3118
H. Base		3		0.3225	-0.0198	0.3027	0.0257	0.3283
			38	0.0250	0.0783	0.1033	0.0736	0.1770
H. Base		5		0.1175	0.0388	0.1563	0.0612	0.2176
			16	0.0200	0.0815	0.1015	0.0753	0.1768
H. Base		8		0.0825	0.0281	0.1106	0.0657	0.1763
			8	0.0350	0.0604	0.0954	0.0772	0.1725
H. Base	5	1		0.4875	-0.0011	0.4764	-0.0178	0.4586
			+997	0.1475	0.1004	0.2479	0.0334	0.2813
H. Base		3		0.3875	-0.0023	0.3852	0.0106	0.3958
			200	0.0950	0.1030	0.1980	0.0452	0.2432
H. Base		5		0.3325	-0.0050	0.3275	0.0225	0.3501
			152	0.0325	0.0915	0.1240	0.0698	0.1939
H. Base		8		0.1750	0.0215	0.1965	0.0448	0.2412
			9	0.0275	0.0854	0.1129	0.0708	0.1837
H. Base	10	(9)		0.1050	0.0492	0.1542	0.0512	0.2055
			20	0.0150	0.0966	0.1116	0.0692	0.1808
RBF	2	(30)		0.0125	0.1064	0.1189	0.0794	0.1982
RBF	5	(15)		0.0650	0.0444	0.1094	0.0679	0.1774

Figure 4: Sesgo-varianza de AdaBoost con redes RBF-OLS. Se muestra: σ, el número de funciones básicas (NB), la iteración óptima (Topt), y las componentes del riesgo.

3. El número óptimo de iteraciones de AdaBoost es más pequeño cuanto mayor es la complejidad de la hipótesis. Si dicha complejidad se debe a un radio pequeño, el sobreajuste se produce más rápidamente que si se debe al número de funciones básicas.

4. Por tanto, para un mismo valor del parámetro de escala, salvo para valores pequeños, es preferible un número mayor de funciones básicas pues se acelera la iteración óptima sin producir un sobreajuste repentino posterior.

5. Con una o dos funciones básicas, en ningún caso AdaBoost supera la mejor red RBF-OLS.

5. REFERENCIAS

Chen, S., Billings, S. A., Luo. W. (1989). "Orthogonal least squares methods and their applications to non-linear system identification". *Int. J. Control*, 50, 1873-1896.

Domingos, P. (2000). "A Unified Bias-Variance Decomposition and its Applications". *Proc. of the 17th ICML*. Morgan Kauffman, 231-238.

Rätsch G., Onoda, T., Müller, K.-R. (1999). "Regularizing AdaBoost". En *Advances in Neural Information Processing Systems*, 11. MIT Press.

Rätsch G., Onoda, T., Müller, K.-R. (2001). "Soft Margins for AdaBoost". *Machine Learning*, 42, 287-320.

Schapire, R., Freund, Y., Bartlett, Y., Lee, W. (1998). "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods". *The Annals of Statistics*, 26, 1651-1686.