ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Ensamble de clasificadores utilizando ADABOOST

Andrés G. Abad, Ph.D.

Agenda

Combinación de Estimaciones

En la conferencia Predictive Analytics World/Toronto (PAW) 2012

Método	Valor	Diferencia
Real	362	-
Ganador (persona)	352	10
Promedio ($N = 61$)	365	3

http://www.predictiveanalyticsworld.com/

Agenda

Agenda

Problema de aprendizaje supervisado I

Considere $(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_m}, y_m)$ donde $\mathbf{x_i} \in \mathcal{X} \subseteq \mathbb{R}^n$, $y_i \in \mathcal{Y} \subseteq \mathbb{R}$.
Asumimos que existe una función no conocida

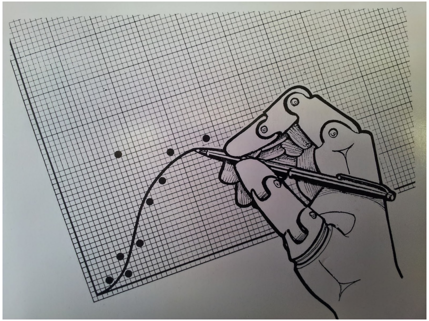
$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

Buscamos una hipótesis

$$h : \mathcal{X} \rightarrow \mathcal{Y}$$

que tenga un bajo error de generalización

$$\epsilon = P[h(\mathbf{x}) \neq f(\mathbf{x})].$$



Problema de aprendizaje supervisado II

Según la naturaleza del conjunto $\mathcal{Y}$ tenemos los siguientes tipos de problemas

$\mathcal{Y}$	Tipo de problema
$\mathbb{R}$	Regresión
$\{c_1, \ldots, c_n\}$	Clasificación
$\{-1, +1\}$	Clasificación binaria

Para problemas de regresión generalmente usamos

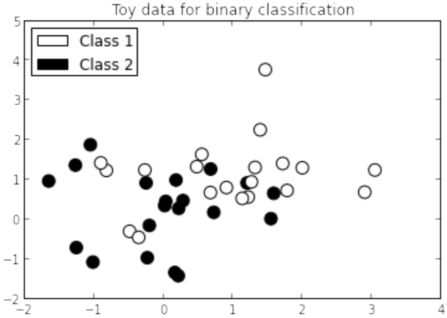
$$\epsilon = MSE(h) = \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} (f(\mathbf{x}) - h(\mathbf{x}))^2$$

Para problemas de clasificación generalmente usamos

$$\epsilon = \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} [\mathbb{I}(h(\mathbf{x}) \neq f(\mathbf{x}))]$$

Métodos de clasificación I

Algunos de los principales algoritmos para clasificación binaria



Clasificación Binaria

- Clasificador bayesiano ingenuo
- Arboles de clasificación (e.g., CART, C4.5)
- Análisis de discriminantes (e.g., lineal, cuadrático)
- Máquinas de Soporte Vectorial
- Redes Neuronales Artificiales
- Regresión logística

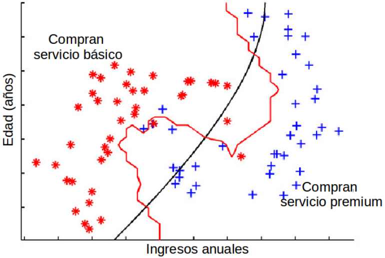
Definición del problema de clasificación I

- Un objeto $\mathbf{x} = [x_1, \ldots, x_p]$, con características x_i, pertenece exactamente a una clases $c \in \{1, 2, \ldots, C\}$.
- Asumimos que tenemos un conjunto de datos

$$\mathcal{D} = \{(\mathbf{x}^{(1)}, c^{(1)}), \ldots, (\mathbf{x}^{(n)}, c^{(n)})\}$$

- Buscamos una función $\hat{f}$ que asigne $\mathbf{x}^{(i)}$ a $c^{(i)}$ lo mejor posible:

$$\hat{f} = \arg\min_{f} \mathbb{P}_{(\mathbf{x},c)}[\mathbb{1}(f(\mathbf{x}) \neq c)]$$



- Objeto $\mathbf{x}$ pertenece a una de dos clases: $\{Basico, Premium\}$
- Objeto $\mathbf{x}$ medidos en dos características: x_1 ingresos anuales, y x_2 edad en años
- Dos clasificadores $\hat{f}$'s: convexo-cuadrático (linea negra) y no-convexo (linea roja)

Clasificador bayesiano ingenuo I

Considera el criterio de *maximo a posteriori* (MAP)

$$c = \arg\max_{c_j \in C} P(x_1, \ldots, x_n | c_j) P(c_j).$$

Bajo el supuesto de independencia entre variables

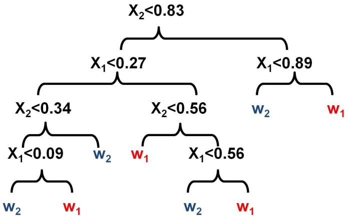
$$c = \arg\max_{c_j \in C} P(c_j) \prod_{i=1}^{n} P(x_i | c_j).$$

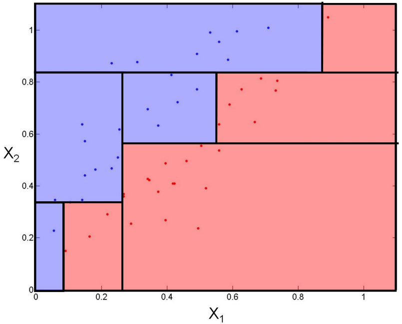
- No considera interacciones entre variables
- No sufre de la maldición de la dimensionalidad
- Si la clase correcta tiene probabilidad alta es robusto al supuesto de independencia

Árboles de clasificación I

Basado en reglas del tipo: Si $A_1 \wedge \cdots \wedge A_m$ entonces c_j

- Generalmente condición A_l de la forma $x_i \geq \theta$





- Algoritmos ID3 [Quinlan, 1986] y C4.5 [Quinlan, 1993] utilizan $H(S) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$
- Algortimo CART utiliza Impureza Gini: $I_G(x) = \sum_{i=1}^{m} x_i(1 - x_i)$

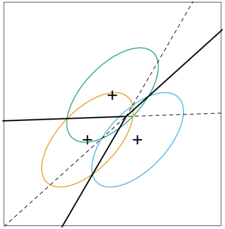
Análisis de discriminante lineal I

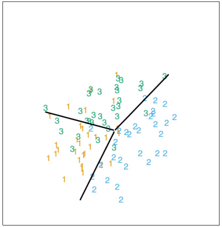
Modelamos la densidad de cada clase con una gaussiana multivariada

$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma_k}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_\mathbf{k})^T \mathbf{\Sigma_k^{-1}}(\mathbf{x}-\mu_\mathbf{k})\right).$$

Asumiremos que las clases tienen matriz de covarianzas común $\mathbf{\Sigma_k} = \mathbf{\Sigma}$

$$\delta_k(\mathbf{x}) = \mathbf{x}^T \mathbf{\Sigma^{-1}} \mu_\mathbf{k} - \frac{1}{2}\mu_\mathbf{k}{}^T \mathbf{\Sigma^{-1}} \mu_\mathbf{k} + \log \pi_k$$





Agenda

Promediando regresiones

Considere

$$H(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} h_i(\mathbf{x}).$$

Se tiene que

$$\begin{aligned} MSE(H) &\leq \overline{MSE}(h) \\ \int \left(\frac{1}{T}\sum_{t=1}^{T} \epsilon_i(\mathbf{x}) \right)^2 p(\mathbf{x}) d\mathbf{x} &\leq \frac{1}{T}\sum_{t=1}^{T} \int \epsilon_i(\mathbf{x})^2 p(\mathbf{x}) d\mathbf{x}, \end{aligned}$$

donde $h_i(\mathbf{x}) = f(\mathbf{x}) + \epsilon_i(\mathbf{x})$ para $i = 1, \ldots, T$.

Si

$$\int \epsilon_i(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = 0 \text{ y } \int \epsilon_i(\mathbf{x}) \epsilon_j(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = 0 \quad (i \neq j)$$

tenemos

$$MSE(H) = \frac{1}{T}\overline{MSE}(h)$$

Sistema de Votación Mayoría Absoluta I

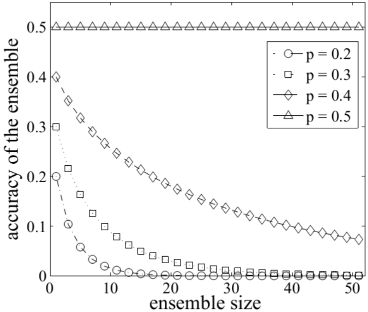
Para el problema de clasificación definimos el ensemble

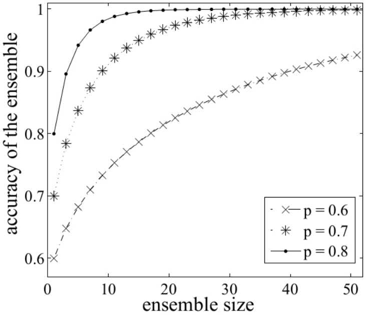
$$H(\mathbf{x}) = \begin{cases} c_j & \text{si } \sum_{i=1}^{T} h_i^j(\mathbf{x}) > \frac{1}{2}\sum_{k=1}^{l}\sum_{i=1}^{T} h_i^k(\mathbf{x}) \\ \text{Rechazo} & \text{si no.} \end{cases}$$

Si asumimos que los clasificadores son independientes y su precisión individual es p tenemos la precisión del ensemble dada por

$$P_{mv} = \sum_{k=\lfloor T/2+1 \rfloor}^{T} \binom{T}{k} p^k (1-p)^{T-k}.$$

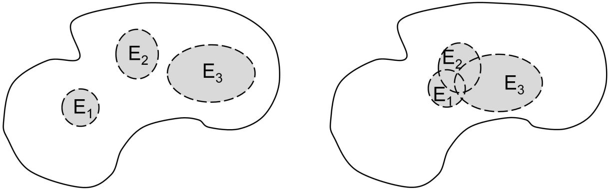
Sistema de Votación Mayoría Absoluta II





- Si $p > 0{,}5$ entonces $\lim_{T\to\infty} P_{mv} = 1$
- Si $p < 0{,}5$ entonces $\lim_{T\to\infty} P_{mv} = 0$
- Si $p = 0{,}5$ entonces $P_{mv} = 0{,}5$ para cualquier T

Diversidad I



A través de las siguientes dos descomposiciones del error cuadrático medio de un ensemble $MSE(H)$

- Descomposición Error-Ambiguedad [Krogh and Vedelsby, 1995]
- Descomposición Sesgo-Varianza-Covarianza [Ueda and Nakano, 1996]

Ambas dependen de un término relacionado con la *diversidad* de los clasificadores.

Descomposición Error-Ambiguedad I

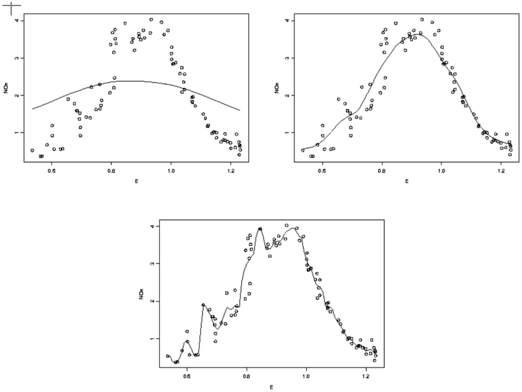
Se puede demostrar que

$$MSE(H) = \overline{MSE}(h) - \overline{AMBI}(h)$$

donde

$$\begin{aligned}
\overline{MSE}(h) &= \int \sum_{i=1}^{T} w_i MSE(h_i|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \\
\overline{AMBI}(h) &= \int \sum_{i=1}^{T} w_i AMBI(h_i|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \\
&= \int \sum_{i=1}^{T} w_i (h_i(\mathbf{x}) - H(\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x}
\end{aligned}$$

Descomposición Sesgo-Varianza-Covarianza I



$$
\begin{aligned}
MSE(h) &= sesgo(h)^2 + var(h) \\
\mathbb{E}\{[h - \mathbb{E}(f)]^2\} &= [\mathbb{E}(h) - \mathbb{E}(f)]^2 + \mathbb{E}\{[h - \mathbb{E}(h)]^2\}
\end{aligned}
$$

Descomposición Sesgo-Varianza-Covarianza II

Así mismo, se puede demostrar que

$$MSE(H) = \overline{SESGO}(H)^2 + \frac{1}{T}\overline{VAR}(H) + \left(1 - \frac{1}{T}\right)\overline{COV}(H)$$

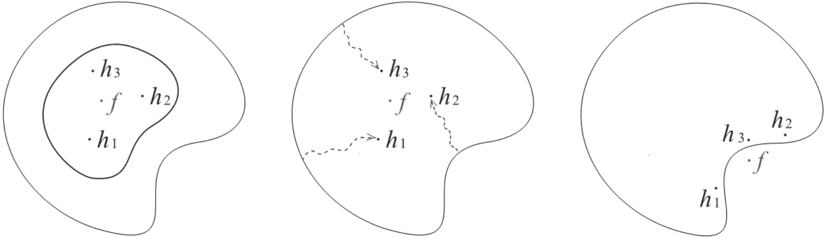
donde

$$\begin{aligned}
\overline{SESGO}(H) &= \frac{1}{T}\sum_{i=1}^{T}(\mathbb{E}[h_i] - f) \\
\overline{VAR}(H) &= \frac{1}{T}\sum_{i=1}^{T}\mathbb{E}(h_i - \mathbb{E}[h_i])^2 \\
\overline{COV}(H) &= \frac{1}{T(T-1)}\sum_{i=1}^{T}\sum_{j=1;j\neq i}^{T}\mathbb{E}(h_i - \mathbb{E}[h_i])\mathbb{E}(h_j - \mathbb{E}[h_j]).
\end{aligned}$$

Métodos para Introducir Diversidad I

- Utilizar un conjunto de datos de entrenamiento de alguna manera diferente
- Seleccionar un subconjunto diferente de variables para entrenar a la hipótesis
- Manipular las etiquetas de las clases
- Introducir aleatoriedad en el algoritmo

Beneficios de combinar hipótesis I



Fuente: [Dietterich, 2000a]

- Problema estadístico
- Problema computacional
- Problema representacional

Principales métodos de ensembles I

Principales métodos de ensembles:

- Clasificador Bayesiano Óptimo
- Bagging (bootstrap aggregating)
  - Random forest
- Boosting
  - AdaBoost (adaptive boosting)

Clasificador Bayesiano Óptimo I

Consideramos $\mathcal{H}$ como el espacio de todas las hipótesis y D una muestra

$$c = \arg\max_{c_j \in C} \sum_{h_i \in \mathcal{H}} P(c_j|h_i)P(h_i|D)$$

Es el mejor clasificador en promedio considerando $\mathcal{H}$ y conocimiento *a priori*

Dificultades prácticas

- $\mathcal{H}$ generalmente muy grande como para iterar
- Hipótesis h generalmente entregan clase y no probabilidades $P(c|h)$
- Calcular probabilidades posterior $P(h|D)$ es generalmente no trivial
  - Necesitamos $P(D|h)$ y $P(h)$

Bagging I

El Bagging (Bootstrap AGGregatING) fue introducido en Breiman [1996]

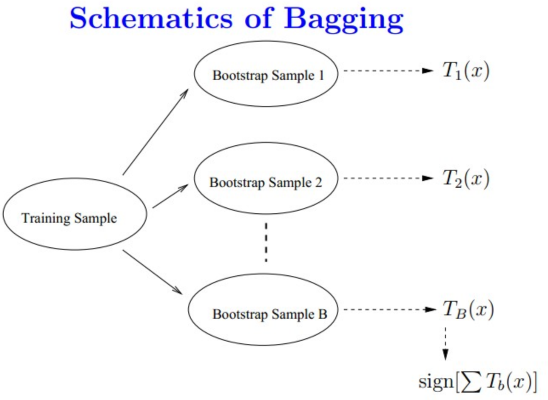
Considere que tenemos

$$\mathcal{L} = \{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_m}, y_m)\}$$

1. Utilizando muestreo aleatorio con reposición y obtenemos

   $$\mathcal{L}_b = \{(\mathbf{x_{b_1}}, y_{b_1}), \ldots, (\mathbf{x_{b_m}}, y_{b_m})\},$$

   para $b = 1, \ldots, B$.
2. Aprendemos h_b utilizando $\mathcal{L}_b$
3. Agregamos hipótesis



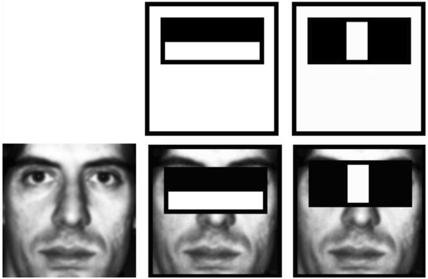
Boosting I

- En Kearns and Valiant [1989] se plantea la pregunta de si las clases de complejidad: aprendedores débiles y aprendedores fuertes, son iguales
- Schapire [1990] responde a esa pregunta, su prueba es constructiva: Boosting

Boosting II

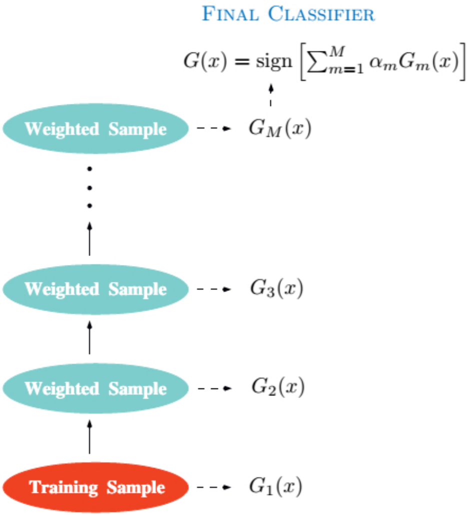
Suponga que $h_1, \ldots, h_T$ son **clasificadores débiles** utilizados para aproximar una función $f : \mathbb{R}^k \to \{-1, +1\}$, tal que

$$\varepsilon = P[h(\mathbf{x}) \neq f(\mathbf{x})] = 0{,}5 - \gamma \qquad \text{para } \mathbf{x} \in \mathcal{X}; \gamma > 0$$



Clasificadores Débiles ([Viola and Jones, 2001])

Boosting III



Agenda

Introducción al AdaBoost I

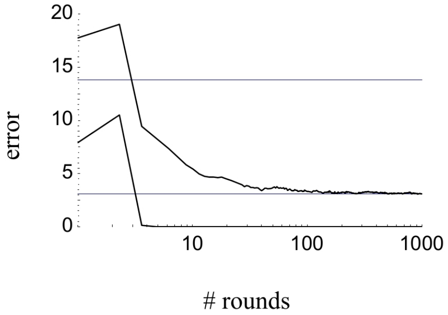
- Basados en Schapire [1990], se introduce en Freund and Schapire [1996] el algoritmo AdaBoost (ADAptive BOOSTing)
- En Freund and Schapire [1997] se realiza la primera extensión del AdaBoost al problema de regresión

Reducción del error en AdaBoost I

Sea $\epsilon_t = \frac{1}{2} - \gamma_t$ el error de entrenamiento de h_t, entonces se puede demostrar que

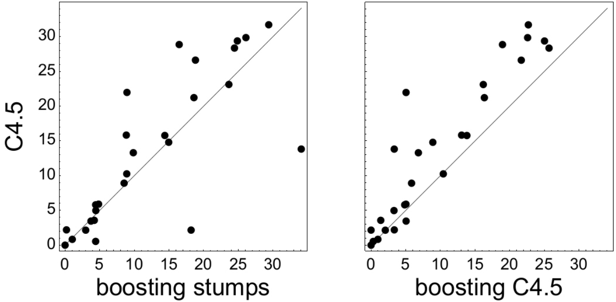
$$\epsilon_H = \prod_t \left[2\sqrt{(\epsilon_t(1-\epsilon_t)}\right]$$

$$\leq \exp\left(-2\sum_t \gamma_t^2\right)$$



Reducción del error en AdaBoost II

Empiricamente se ha evidenciado la superioridad del AdaBoost [Freund and Schapire, 1996; Bauer and Kohavi, 1999; Dietterich, 2000b]



Comparación de error de prueba entre algoritmos C4.5 Vs. Boosting Decision Stumps, y Boosting C4.5 respectivamente [Freund and Schapire, 1999].

Descripción general I

El AdaBoost es una forma de optimización gradiente en el espacio de hipótesis con el objetivo de minimizar la **función de pérdida exponencial**

$$\ell_{\text{exp}}(f, H|\mathcal{D}) = \mathbb{E}_{x\sim\mathcal{D}}[e^{-f(x)H(x)}]$$

para

$$H(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

Descripción general II

Al minimizar la función de pérdida exponencial $\ell_{\text{exp}}(f, H|\mathcal{D})$ tenemos

$$\begin{aligned}\frac{\partial e^{-f(\mathbf{x})H(\mathbf{x})}}{\partial H(\mathbf{x})} &= -f(\mathbf{x})e^{-f(\mathbf{x})H(\mathbf{x})} \\ &= e^{-H(\mathbf{x})}P(f(\mathbf{x}) = +1|\mathbf{x}) + e^{H(\mathbf{x})}P(f(\mathbf{x}) = -1|\mathbf{x}) = 0\end{aligned}$$

Resolviendo

$$H(\mathbf{x}) = \frac{1}{2}\ln\frac{P(f(\mathbf{x}) = +1|\mathbf{x})}{P(f(\mathbf{x}) = -1|\mathbf{x})}$$

Descripción general III

Dado que

$$
\begin{aligned}
\text{sign}\,(H(\mathbf{x})) &= \text{sign}\left(\frac{1}{2}\ln\frac{P(f(\mathbf{x})=+1|\mathbf{x})}{P(f(\mathbf{x})=-1|\mathbf{x})}\right) \\
&= \begin{cases} 1 & \text{si } P(f(\mathbf{x})=+1|\mathbf{x}) > P(f(\mathbf{x})=-1|\mathbf{x}); \\ -1 & \text{si } P(f(\mathbf{x})=+1|\mathbf{x}) < P(f(\mathbf{x})=-1|\mathbf{x}) \end{cases} \\
&= \arg\max_{y\in\{-1,+1\}} P(f(\mathbf{x})=y|\mathbf{x})
\end{aligned}
$$

lo que implica que $\text{sign}\,(H(\mathbf{x}))$ alcanza la tasa de error bayesiano.

Descripción general IV

Para $t = 1, \ldots, T$:

1. Entrenar la hipótesis débil $h_t : \mathcal{X} \to \{-1, +1\}$ utilizando la distribución $\mathcal{D}_t$

Obtener $H(\mathbf{x}) = \sum_{i=1}^{T} \alpha_i h_i(\mathbf{x})$.

Para completamente definir el AdaBoost necesitamos definir

- Como determinar las distribuciones $\mathcal{D}_t$
- Cómo determinar los pesos α_t

Descripción general V

El clasificador h_t que corrige los errores de H_{t-1} debe minimizar la función de pérdida exponencial

$$
\begin{aligned}
\ell_{exp}(H_{t-1}+h_t|\mathcal{D}) &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[e^{-f(\mathbf{x})(H_{t-1}(\mathbf{x})+h_t(\mathbf{x}))}\right] \\
&\approx \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[e^{-f(\mathbf{x})H_{t-1}(\mathbf{x})}\left(1-f(\mathbf{x})h_t(\mathbf{x})+\frac{f(\mathbf{x})^2h_t(\mathbf{x})^2}{2}\right)\right] \\
&= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[e^{-f(\mathbf{x})H_{t-1}(\mathbf{x})}\left(1-f(\mathbf{x})h_t(\mathbf{x})+\frac{1}{2}\right)\right]
\end{aligned}
$$

Descripción general VI

El clasificador ideal h_t sera tal que

$$\begin{aligned}
h_t(\mathbf{x}) &= \arg\min_h \ell_{exp}(H_{t-1} + h|\mathcal{D}) \\
&\approx \arg\min_h \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[e^{-f(\mathbf{x})H_{t-1}(\mathbf{x})}\left(1 - f(\mathbf{x})h(\mathbf{x}) + \frac{f(\mathbf{x})^2 h(\mathbf{x})^2}{2}\right)\right] \\
&= \arg\max_h \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[e^{-f(\mathbf{x})H_{t-1}(\mathbf{x})} f(\mathbf{x})h(\mathbf{x})\right] \\
&= \arg\max_h \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\frac{e^{-f(\mathbf{x})H_{t-1}(\mathbf{x})}}{\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[e^{-f(\mathbf{x})H_{t-1}(\mathbf{x})}]} f(\mathbf{x})h(\mathbf{x})\right] \\
&= \arg\max_h \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t}[f(\mathbf{x})h(\mathbf{x})] \\
&= \arg\min_h \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t}[\mathbb{I}(f(\mathbf{x}) \neq h(\mathbf{x}))]
\end{aligned}$$

para $\mathcal{D}_t(\mathbf{x}) = \frac{\mathcal{D}(\mathbf{x})e^{-f(\mathbf{x})H_{t-1}(\mathbf{x})}}{\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[e^{-f(\mathbf{x})H_{t-1}(\mathbf{x})}]}$.

Descripción general VII

Bajo una distribución $\mathcal{D}_t$, el peso α_t se escoge minimizando la función de pérdida exponencial

$$
\begin{aligned}
\ell_{\text{exp}}(f, \alpha_t h_t | \mathcal{D}_t) =& \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_t}\left[e^{-f(\mathbf{x})\alpha_t h_t(\mathbf{x})}\right] \\
=& \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_t}\left[e^{-\alpha_t} \mathbb{I}(f(\mathbf{x}) = h_t(\mathbf{x})) + e^{\alpha_t} \mathbb{I}(f(\mathbf{x}) \neq h_t(\mathbf{x}))\right] \\
=& e^{-\alpha_t} P_{\mathbf{x} \sim \mathcal{D}_t}(f(\mathbf{x}) = h_t(\mathbf{x})) + e^{\alpha_t} P_{\mathbf{x} \sim \mathcal{D}_t}(f(\mathbf{x}) \neq h_t(\mathbf{x})) \\
=& e^{-\alpha_t}(1 - \epsilon_t) + e^{\alpha_t} \epsilon_t
\end{aligned}
$$

donde $\epsilon_t = P_{\mathbf{x} \sim \mathcal{D}_t}(f(\mathbf{x}) \neq h_t(\mathbf{x}))$.

Descripción general VIII

Para obtener el α_t óptimo hacemos

$$\frac{\partial \ell_{\text{exp}}(f, \alpha_t h_t | \mathcal{D}_t)}{\partial \alpha_t} = -e^{-\alpha_t}(1 - \epsilon_t) + e^{\alpha_t}\epsilon_t = 0$$

cuya solución es

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

Algoritmo AdaBoost I

Inicialice: $\mathcal{D}_1(i) = 1/m$ para $i = 1, \ldots, m$.
Para $t = 1, \ldots, T$:

1. Entrenar la hipótesis débil $h_t : \mathcal{X} \to \{-1, +1\}$ utilizando la distribución $\mathcal{D}_t$
2. Evalue error ponderado:

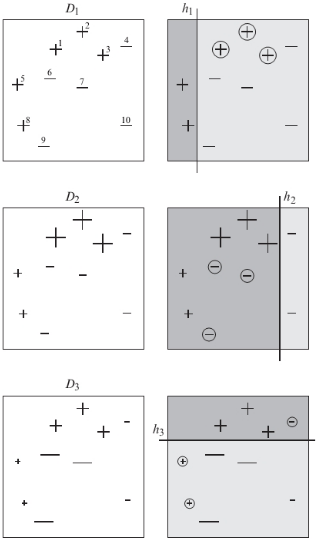
$$\epsilon_t = Pr_{i \sim \mathcal{D}_t}[h_t(x_i) \neq y_i]$$

3. Seleccione $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
4. Actualice para $i = 1, \ldots, m$:

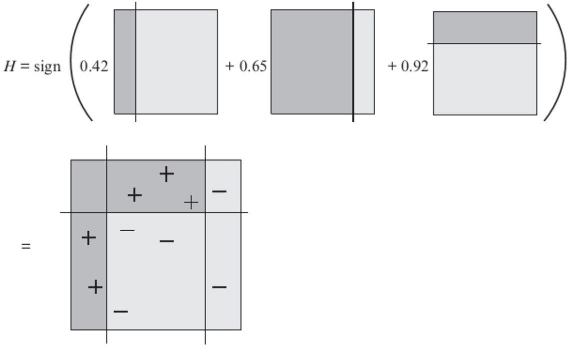
$$\mathcal{D}_{t+1}(i) = \frac{\mathcal{D}_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t},$$

donde Z_t es el factor de normalización

Algoritmo AdaBoost II



$$\alpha_1 = 0{,}42, \alpha_2 = 0{,}65, \alpha_3 = 0{,}92$$



Agenda

Conclusiones I

- Los métodos de ensembles reducen el error de entrenamiento y el de prueba
- El concepto de diversidad entre hipótesis es central
  - Existen diferentes maneras de introducir diversidad a las hipótesis
- El AdaBoost es un algoritmo específico para el Boosting que introduce diversidad ajustando la distribución de la muestra
  - El Boosting reduce asintóticamente el error de entrenamiento exponencialmente

Referencias Bibliográficas I

Bauer, E. and Kohavi, R. (1999). An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, 36(1-2):105–139.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Dietterich, T. G. (2000a). Ensemble Methods in Machine Learning. In *Multiple Classifier Systems*, number 1857 in Lecture Notes in Computer Science, pages 1–15. Springer Berlin Heidelberg.

Dietterich, T. G. (2000b). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Machine Learning*, 40(2):139–157.

Freund, Y. and Schapire, R. (1996). Experiments with a New Boosting Algorithm. pages 148–156.

Freund, Y. and Schapire, R. (1999). A short introduction to boosting. *Japonese Society for Artificial Intelligence*, 14(5):771–780.

Freund, Y. and Schapire, R. E. (1997). *A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting*.

Kearns, M. and Valiant, L. (1989). *Cryptographic Limitations on Learning Boolean Formulae and Finite Automata*.

Referencias Bibliográficas II

Krogh, A. and Vedelsby, J. (1995). Neural Network Ensembles, Cross Validation, and Active Learning. In *Advances in Neural Information Processing Systems*, pages 231–238. MIT Press.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 1(1):81–106.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5(2):197–227.

Ueda, N. and Nakano, R. (1996). Generalization error of ensemble estimators. In *, IEEE International Conference on Neural Networks, 1996*, volume 1, pages 90–95 vol.1.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001*, volume 1, pages I–511–I–518 vol.1.

Apéndice I

Una expansión aditiva de funciones bases toma la forma

$$f(\mathbf{x}) = \sum_{m=1}^{M} \beta_m b(\mathbf{x}; \gamma_m)$$

El ajuste se realiza minimizando función de pérdida

$$\min_{\{\beta_m, \gamma_m\}_1^M} \sum_{i=1}^{N} L\left(y_i, \sum_{m=1}^{M} \beta_m b(\mathbf{x}_i; \gamma_m)\right)$$

$$\min_{\beta, \gamma} \sum_{i=1}^{N} L\left(y_i, \beta b(\mathbf{x}_i; \gamma)\right)$$

Apéndice II

Algoritmo: Ajuste por Etapas hacia Adelante

1. Inicialice $f_0(\mathbf{x}) = 0$
2. Para $m = 1, \ldots, M$:
   a Calcule

$$(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, f_{m-1}(\mathbf{x}_i) + \beta b(\mathbf{x}_i; \gamma))$$

   b Establezca $f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \beta_m b(\mathbf{x}; \gamma_m)$