EE E6887 (Statistical Pattern Recognition)
Solutions for homework 7

P.1 One of the "conservation laws" for generalization states that the positive generalization performance of an algorithm in some learning situations must be offset by negative performance elsewhere. Consider a very simple learning algorithm that seems to contradict this law. For each test pattern, the prediction of the majority learning algorithm is merely the category most prevalent in the training data.

(a) Show that averaged over all two-category problems of a given number of features the off-training set error is 0.5.

(b) Repeat (a) but for the minority learning algorithm, which always predicts the category label of the category least prevalent in the training data.

(c) Use your answers from (a) and (b) to illustrate Part (2) of the No Free Lunch Theorem.

Answer:

(a) Let ω_1 and ω_2 represent the two categories. $P(\omega_1)$ and $P(\omega_2) = 1 - P(\omega_1)$ are, respectively, the probabilities that a randomly selected training sample come from ω_1 and ω_2. For a new testing sample $\mathbf{x}$, the true probability that it comes from ω_1 is also $P(\omega_1)$.

Now let's first fix the training set. That is, without loss of generality, we assume that with a particular fixed training set $\mathcal{D}$, the majority learning algorithm predict every test sample $\mathbf{x}$ as from ω_1. Then the expected error over all the two-category problem will be:

$$\mathcal{E}(E|\mathcal{D}) = \int_0^1 (1 - P(\omega_1)) dP(\omega_1) = 0.5$$

Then lets consider all the possible training sets. The error averaged over all the training set will be:

$$\mathcal{E} = \sum_{\mathcal{D}} \mathcal{E}(E|\mathcal{D}) P(\mathcal{D})$$

Since all the possible training sets are equally probable to occur, we have

$$\mathcal{E} = 0.5$$

(b) Similar to case (a), first let's fix the training set. Without loss of generality, we assume that with a particular fixed training set $\mathcal{D}$, the minority learning algorithm predict every test sample $\mathbf{x}$ as from ω_2. Then the expected error over all the two-category problem will be:

$$\mathcal{E}(E|\mathcal{D}) = \int_0^1 P(\omega_1) dP(\omega_1) = 0.5$$

Then lets consider all the possible training sets. The error averaged over all the training set will be:

$$\mathcal{E} = \sum_{\mathcal{D}} \mathcal{E}(E|\mathcal{D}) P(\mathcal{D})$$

Since all the possible training sets are equally probable to occur, we have

$$\mathcal{E} = 0.5$$

(c) The part 2 of No Free Lunch is: For any fixed training set $\mathcal{D}$, uniformly averaged over F, $\sum_F \mathcal{E}_1(E|F,\mathcal{D}) - \mathcal{E}_2(E|F,\mathcal{D}) = 0$. Case (a) and (b) just gives an example for this. For a fixed training set $\mathcal{D}$, for a particular $P(\omega_1)$, the majority learning and minority learning algorithm may gives out different training error. But when uniformly averaged over all possible target problems, the expected error rate will be the same.

P.2 Consider AdaBoost with an arbitrary number of component classifiers.

(a) State clearly any assumptions you make, and derive Eq. 37 for the ensemble training error of the full boosted system.

(c) Recall that the training error for a weak learner applied to a two-category problem can be written $E_k = 1/2 - G_k$ for some positive value G_k. The training error for the first component classifier is $E_1 = 0.25$. Suppose that G_k decreases as a function of k. Specifically, repeat part (b) with the assumption $G_k = 0.05/k$ for $k = 1$ to k_{max} (Plot the upper bound on the ensemble test error given by Eq.37, such as shown in Fig.9.7.

Answer:

The following proof is outlined in [1]

(a) We can rewrite the weight updating rule as:

$$W_{k+1}(i) = \frac{W_k(i)}{Z_k} \times \exp\{-\alpha_k h_k(\mathbf{x}_i) y_i\}$$

So

$$W_{k_{max}+1}(i) = W_1(i) \left(\prod_{k=1}^{k_{max}} \frac{1}{Z_k} \right) \exp\left\{ \sum_{k=1}^{k_{max}} [-\alpha_k h_k(\mathbf{x}_i) y_i] \right\}$$

If the ensemble hypothesis makes a mistake, $y_i \cdot \sum_{k=1}^{k_{max}} [-\alpha_k h_k(\mathbf{x}_i)] = -1$, so $\exp\{\sum_{k=1}^{k_{max}} [-\alpha_k h_k(\mathbf{x}_i) y_i]\} \geq 1$. Thus we have:

$$\begin{aligned}
\sum_{i:\ h_k(\mathbf{x}_i) \neq y_i} W_1(i) &\leq \sum_{i:\ h_k(\mathbf{x}_i) \neq y_i} W_1(i) \exp\left\{ \sum_{k=1}^{k_{max}} [-\alpha_k h_k(\mathbf{x}_i) y_i] \right\} \\
&= \sum_{i:\ h_k(\mathbf{x}_i) \neq y_i} W_{k_{max}+1}(i) \left(\prod_{k=1}^{k_{max}} Z_k \right) \\
&\leq \left(\sum_{i=1}^{N} W_{k_{max}+1}(i) \right) \left(\prod_{k=1}^{k_{max}} Z_k \right) \\
&= \prod_{k=1}^{k_{max}} Z_k
\end{aligned}$$

That is:

$$E \leq \prod_{k=1}^{k_{max}} Z_k \tag{1}$$

Moreover, since

$$\begin{aligned}
Z_k &= \sum_{i=1}^{N} W_k(i) \exp\{-\alpha_k h_k(\mathbf{x}_i) y_i\} \\
&= \sum_{i=1}^{N} W_k(i) \exp\left\{-\alpha_k(\frac{1+h_k(\mathbf{x}_i)y_i}{2}) + \alpha_k(\frac{1-h_k(\mathbf{x}_i)y_i}{2})\right\} \\
&\leq \sum_{i=1}^{N} W_k(i) \left[(\frac{1+h_k(\mathbf{x}_i)y_i}{2})e^{-\alpha_k} + (\frac{1-h_k(\mathbf{x}_i)y_i}{2})e^{\alpha_k}\right] \\
&= (1-E_k)e^{-\alpha_k} + E_k e^{\alpha_k} \\
&= (1-E_k)\left(\frac{1-E_k}{E_k}\right)^{-1/2} + E_k\left(\frac{1-E_k}{E_k}\right)^{1/2} \\
&= 2\sqrt{E_k(1-E_k)}
\end{aligned}$$

So we get:

$$E \leq \prod_{k=1}^{k_{max}} 2\sqrt{E_k(1-E_k)}$$

Since $E_k = 1/2 - G_k$, we have

$$E \leq \prod_{k=1}^{k_{max}} \sqrt{1-4G_k^2} \leq \exp\left(-2\sum_{k=1}^{k_{max}} G_k^2\right)$$

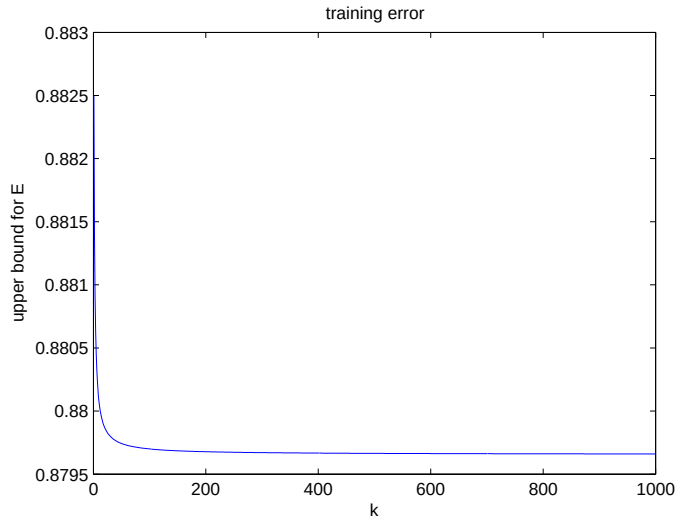
The second inequality is due to the fact that $1+x \leq e^x$ for ($1+x$ is the tangent line of e^x at $x=0$. As e^x is convex, hence $1+x \leq e^x$).

(c) We use the following program to plot the upper bound of the training error.

```
kmax = 1000;
k = 1:kmax;
G = 0.05./k;
G(1) = 0.25;
for i=1:kmax
e(i) = exp(-2*(sum(G(1:i).^ 2)));
end
```

```
plot(e);
xlabel('k');
ylabel('upper bound for E');
title('training error');
```

The result is as follows:



References

[1] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machince Learning*, vol. 37, no. 3, pp. 297–336, 1999.